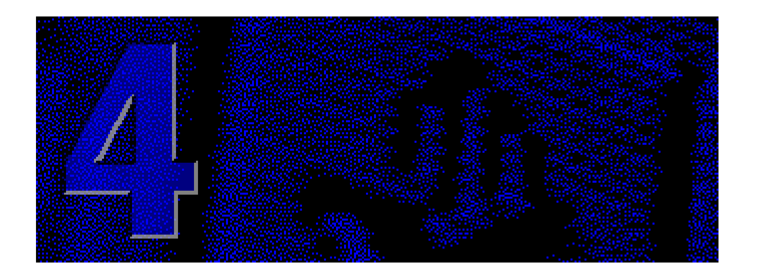
Manual del programador, Parte 4: Agrupar todos los elementos



Ahora ya puede completar su aplicación y probar su funcionalidad. Agregue consultas e informes para ofrecer a los usuarios la información que necesitan y después ejecute la aplicación y busque áreas que pueda optimizar.

Capítulo 12 Agregar consultas e informes

Después de crear datos y una interfaz para su aplicación, puede aumentar su eficacia con consultas e informes que proporcionen información importante para sus usuarios .

Capítulo 13 Compilar una aplicación

Cree sus aplicaciones paso a paso y compruebe cada componente. Cuando haya incluido todos los componentes, será fácil compilarlos en una aplicación.

Capítulo 14 Probar y depurar aplicaciones

Cuando programe una aplicación, necesitará comprobar si existen errores en los componentes. Visual FoxPro ofrece herramientas de depuración para ayudarle a buscar y corregir cualquier error que descubra en sus aplicaciones.

Capítulo 15 Optimizar aplicaciones

Cuando disponga de una aplicación estable y en funcionamiento, considere las formas en que puede optimizar su rendimiento y convertirla en más pequeña y rápida.

Capítulo 12: Agregar consultas e informes

Una vez creadas las tablas y formularios para su aplicación, puede agregar consultas e informes para seleccionar y mostrar datos a los usuarios. Las consultas pueden ir dirigidas hacia diversos destinos, de forma que puede usarlas en los otros componentes de su aplicación. De igual forma, puede ejecutar informes de manera independiente, sin necesidad de utilizar una consulta. Este capítulo resalta algunas de las formas en las que puede utilizar las consultas, agregar informes y exponer consultas e informes a los usuarios.

Cuando utiliza una consulta o una vista en su aplicación, en realidad utiliza una instrucción SELECT - SQL. Este capítulo describe cómo puede utilizar en su aplicación una instrucción SELECT - SQL si la crea mediante una consulta definida en el Diseñador de consultas, una vista definida en el Diseñador de vistas o código introducido para un evento o procedimiento. Para obtener información detallada sobre las vistas, vea la parte 2, Buscar información, del *Manual del usuario*.

Este capítulo trata los temas siguientes:

- Agregar consultas
- Agregar informes y etiquetas
- Integrar consultas e informes

Agregar consultas

Cuando agrega consultas a su aplicación, puede combinar varios orígenes de datos, filtrar registros, manipular datos y ordenar los resultados, todo ello con la instrucción SELECT - SQL. Al utilizar instrucciones SQL, dispone de un control completo sobre los resultados obtenidos en sus consultas y sobre dónde se almacenan estos resultados.

Una consulta es una instrucción SELECT - SQL.



Puede agregar instrucciones SELECT - SQL a procedimientos o código de eventos. Para ver más información acerca de los eventos, consulte el capítulo 4, Descripción del modelo de eventos.

Crear una instrucción SELECT - SQL

Para crear una instrucción SELECT - SQL

- Utilice el Diseñador de consultas o el Diseñador de vistas para crear la instrucción y copie el contenido de la ventana SQL a una ventana de código.

  –O bien–

- En una ventana de código, escriba la instrucción SELECT - SQL.

Por ejemplo, puede seleccionar todos los registros de la tabla `Customer` de la base de datos `TasTrade`

en los que el campo `country` contenga el valor "Canadá":

```
SELECT * ;
   FROM tastrade!customer ;
   WHERE customer.country = "Canadá"
```

Para ejecutar el comando inmediatamente, puede introducir la instrucción en la ventana Comandos. Si desea que cada cláusula aparezca en una línea distinta dentro de la ventana, termine cada línea, excepto la última, con un punto y coma. De esta forma Visual FoxPro procesa el comando sólo después de la última línea.

Seleccionar un número o porcentaje de registros

Si sólo necesita un cierto número o porcentaje de registros del conjunto de resultados que su consulta devolvería, puede usar la propiedad Registros incluidos de la ficha Varios de los Diseñadores de consultas o de vistas o puede agregar una cláusula TOP a su instrucción SELECT - SQL. El número que proporcione en una cláusula TOP puede variar entre 1 y 32767. Para un porcentaje, puede usar un valor entre 0.001 y 99.99.

Por ejemplo, si quiere seleccionar los diez clientes principales con las cantidades de pedidos más altas, puede especificar un GROUP BY en CUST_ID para mostrar un registro agregado para cada cliente y ordenar por ORDER_AMT en la cláusula ORDER BY. Para obtener un TOP 10 verdadero, debe especificar una ordenación descendente en ORDER_AMT de forma que los registros con las cantidades de pedidos más altas aparezcan en primer lugar en el resultado. Si usa un orden ascendente, los registros de resultados se ordenan desde la menor cantidad de pedidos a la mayor. Los registros incluidos que seleccione del conjunto de resultados tendrían en realidad los valores más bajos.

```
SELECT TOP 10 *;
 FROM  testdata!customer INNER JOIN testdata!orders ;
   ON  Customer.cust_id = Orders.cust_id;
GROUP BY Customer.cust_id;
ORDER BY Orders.order_amt DESC
```

Especificar destinos para resultados de consultas

Al utilizar cláusulas de la instrucción SELECT - SQL, puede especificar varios destinos para almacenar el resultado de sus consultas.

Para enviar los resultados a este destino	Utilice esta cláusula
Otra tabla	INTO TABLE mitabla
Matriz	INTO ARRAY aMiMatriz
Tabla temporal	INTO CURSOR micursor
Ventana activa	TO SCREEN
Ventana Examinar	Predeterminado si no se especifica otro destino.

Una vez que los resultados están almacenados, puede utilizar comandos para controlar cómo se integran los resultados almacenados para su presentación o impresión.

Almacenar los resultados en una tabla, matriz o cursor

Puede almacenar los resultados de sus consultas en una tabla, una matriz o un cursor para otros usos, tales como completar formularios e imprimir informes y etiquetas. Si desea almacenar los resultados sólo temporalmente, envíelos a una matriz o a un cursor. Si lo que desea es almacenar los resultados definitivamente, envíelos a una tabla.

Para especificar una tabla como destino

- Utilice la cláusula INTO de la instrucción SELECT - SQL para especificar un destino.

Este ejemplo muestra una cláusula INTO para una tabla:

```
SELECT * ;
   FROM tastrade!customer ;
   WHERE customer.country = "Canadá" ;
   INTO TABLE mitabla
```

Para especificar una matriz como destino

- Utilice la cláusula INTO de la instrucción SELECT - SQL para especificar un destino.

El ejemplo siguiente muestra una cláusula INTO para una matriz:

```
SELECT * ;
   FROM tastrade!customer ;
   WHERE customer.country = "Canadá" ;
   INTO ARRAY aMiMatriz
```

Para especificar un cursor como destino

- Utilice la cláusula INTO de la instrucción SELECT - SQL para especificar un destino.

Este ejemplo muestra una cláusula INTO para un cursor llamado `micursor`.

```
SELECT * ;
   FROM tastrade!customer ;
   WHERE customer.country = "Canadá" ;
   INTO CURSOR micursor
```

Si crea una tabla o una matriz, puede utilizarla como cualquier otra tabla o matriz en Visual FoxPro. Si crea un cursor, puede examinar o anexar su contenido. El cursor se abre en el menor área de trabajo disponible. Puede tener acceso al mismo con el nombre que le ha dado en la instrucción SELECT - SQL.

Los dos procedimientos siguientes describen dos formas comunes para incluir en una aplicación los resultados de consultas almacenados en tablas y cursores.

Rellenar un control de formulario

Si desea mostrar los resultados de sus consultas en un formulario, puede utilizar una tabla, una matriz o un cursor para colocarlos en una cuadrícula, un cuadro de lista o un cuadro combinado.

Para rellenar un control cuadro de lista o cuadro combinado con una tabla o cursor

1. En el Diseñador de formularios, modifique el formulario que tiene el control que desea rellenar.
2. Establezca la propiedad RowSourceType a **3 - SQL Statement**.
3. En la propiedad RowSource del control, escriba una instrucción SELECT - SQL que incluya una cláusula INTO TABLE o INTO CURSOR.

Para llenar un control cuadrícula con una tabla o un cursor

1. En el Diseñador de formularios, modifique el formulario que tiene el control que desea llenar.
2. En el evento Load del formulario, escriba una instrucción SELECT - SQL que incluya una cláusula INTO TABLE o INTO CURSOR.
3. Establezca la propiedad RecordSource de la cuadrícula como el nombre de la tabla o cursor que haya creado en el paso 2.
4. Establezca la propiedad RecordSourceType de la cuadrícula a **0 – Table** (para una tabla) o **1 – Alias** (para un cursor).

Imprimir el resultado en un informe o una etiqueta

Si su informe o etiqueta incluye grupos, o si necesita ordenar los datos de alguna otra manera, puede utilizar las distintas cláusulas de la instrucción SELECT - SQL para obtener el resultado exacto que necesita.

Para enviar los resultados a un informe o una etiqueta existente

- Utilice la instrucción SELECT - SQL con un comando REPORT o LABEL.

  El ejemplo siguiente utiliza las cláusulas GROUP BY y ORDER BY, así como el comando REPORT FORM:

```
SELECT * ;
FROM tastrade!customer ;
WHERE customer.country = "Canadá" ;
GROUP BY customer.region ;
ORDER BY customer.postal_code, customer.company_name ;
INTO CURSOR MiCursor
REPORT FORM MYREPORT.FRX
```

El ejemplo siguiente utiliza un comando LABEL FORM:

```
SELECT * ;
FROM tastrade!customer ;
WHERE customer.country = "Canadá" ;
GROUP BY customer.region ;
ORDER BY customer.postal_code, customer.company_name ;
INTO CURSOR micursor
LABEL FORM MYLABEL.LBX
```

Aunque la instrucción SELECT - SQL es el método más flexible para rellenar sus informes o etiquetas, no es el único método. Para obtener más información acerca de cómo establecer orígenes de datos para informes, consulte la sección Controlar los orígenes de datos de un informe, más adelante en este mismo capítulo. Para ver más información acerca de la integración de los destinos de informes en sus aplicaciones, consulte la sección Integrar consultas e informes, más adelante en este mismo capítulo.

Mostrar el resultado en una ventana

Si desea mostrar el resultado de su instrucción SELECT - SQL, puede enviarlo a una ventana. La ventana Examinar es el destino predeterminado para los resultados de las consultas y no necesita incluir una cláusula de destino. Puede enviar también el resultado a la ventana principal de Visual FoxPro o a otra ventana activa.

Para mostrar resultados en la ventana principal de Visual FoxPro

- Utilice la cláusula TO SCREEN de una instrucción SELECT - SQL.

Para mostrar resultados en otra ventana activa

- Defina una ventana, muéstrela para activarla y, a continuación, ejecute una consulta SQL u otro comando que muestre resultados en una ventana.

Este programa de ejemplo muestra la definición para una ventana temporal titulada Principales clientes, que muestra los nombres de las compañías con una facturación anual superior a los $50.000.

Mostrar los resultados de la consulta en una ventana

Código	Comentario

`frmMyForm=createobj("form")`
`frmMyForm.Left = 1`
`frmMyForm.Top = 1`
`frmMyForm.Width = 130`
`frmMyForm.Height = 25`
`frmMyForm.Caption = "Principales clientes"`
`frmMyForm.Show`
	Crea e inicia un objeto ventana temporal.

`SELECT customer.company_name,`
`  SUM(orders.freight) ;`
`  FROM tastrade!customer,`
	Introduce una instrucción SELECT - SQL.

```
tastrade!orders ;
WHERE customer.customer_id =
orders.customer_id ;
GROUP BY customer.company_name ;
HAVING SUM(orders.freight) > 5000 ;
ORDER BY 2 DESC
```

Agregar informes y etiquetas

Después de recoger y organizar sus datos, puede agregar informes o etiquetas a su aplicación para imprimir los datos o mostrarlos en pantalla. Puede controlar los datos de su informe mediante los orígenes de datos que elija o manipular y combinar datos sin manipular con *variables de informe*. Las variables de informe almacenan valores que se calculan y utilizan en un informe.

Controlar los orígenes de datos

Para controlar los orígenes de datos de un informe puede definir un entorno de datos que se almacenan con el informe o bien puede, mediante código, activar orígenes de datos específicos cada vez que ejecute un informe. Para ver más información acerca del uso del Diseñador de entornos de datos, consulte el capítulo 9, Crear formularios.

Para	Agregue
Utilizar siempre los mismos orígenes de datos.	Tablas o útiles al entorno de datos del informe.
	DO *consulta* o SELECT - SQL al código de evento Init del entorno de datos del informe.
Utilizar conjuntos diferentes de orígenes de datos.	USE *tabla*, USE *vista*, DO *consulta* o SELECT - SQL para el evento Click u otro código que preceda a un comando REPORT o LABEL.

Si utiliza una tabla como origen de datos, los registros se procesarán e imprimirán en el orden en que aparecen en la tabla. Utilice alias sólo si no piensa usar el informe con ningún otro origen de datos que no sea la propia tabla. Si utiliza una vista o una consulta como origen de datos y los alias están incluidos en los controles del informe, éste podría mostrar de forma repetida el mismo registro en la página.

Controlar el orden de registros

Puede usar los orígenes de datos usados por el informe para controlar el orden en que se imprimen los registros en su informe. Los registros se imprimen en el orden en que aparecen en la tabla, vista o consulta. Para ordenar los registros de una tabla, puede establecer un índice en el código o como parte del entorno de datos de un informe. Para una consulta, vista o código SELECT - SQL, puede usar la cláusula ORDER BY. Si no ordena los registros mediante los orígenes de datos, la única forma de usar únicamente el informe para ordenar los registros es a través de la propiedad ORDER de un cursor del entorno de datos.

Controlar la selección de registros

Además del orden en que los registros aparecen en el informe, puede seleccionar qué registros se imprimen con el origen de datos, las opciones de impresión de informes o una combinación de ambas cosas.

Para usar	Agregue
Vista o consulta	Condiciones a la ficha Filtro
SELECT - SQL	Cláusula WHERE o HAVING
Diseñador de informes	Configuración en el cuadro de diálogo Opciones de impresión
Comando REPORT	Expresiones Alcance, FOR o WHILE
Tabla	Índice filtro

Proteger una sesión de datos de un informe

Para evitar que la sesión de datos de su informe se vea afectada por la sesión de datos global como resultado de las modificaciones realizadas por otros diseñadores, puede establecer la sesión de datos del informe como privada.

Para establecer una sesión privada de datos

- En el menú **Informe**, elija **Sesión privada de datos**.

Para obtener más información sobre el uso del Diseñador de entornos de datos, consulte el capítulo 9, Crear formularios. Para obtener más información acerca de las sesiones de datos, consulte el capítulo 17, Programar para acceso compartido.

Si desea mostrar el resultado de una consulta en un gráfico, puede usar el Asistente para gráficos, el Diseñador de consultas o un comando SELECT - SQL. Para usar el Diseñador de consultas o un comando SELECT - SQL, siga los pasos que se indican a continuación. Debe incluir por lo menos un campo numérico en el conjunto de resultados para crear un gráfico. Después de terminar la consulta, puede elegir entre seis tipos de gráficos, cada uno con dos variantes.

Para modificar el gráfico

1. Examine la tabla que contiene el gráfico.
2. Haga doble clic en el campo general para mostrar el gráfico.
3. Haga doble clic en el gráfico para abrir Microsoft Graph y mostrar la barra de herramientas de Microsoft Graph.
4. Modifique el gráfico en Microsoft Graph.

Refinar el diseño de página

Puede refinar el diseño de las páginas de su informe; para ello, defina múltiples columnas y cambie el área de página reservada para una banda cambiando el alto de cada banda.

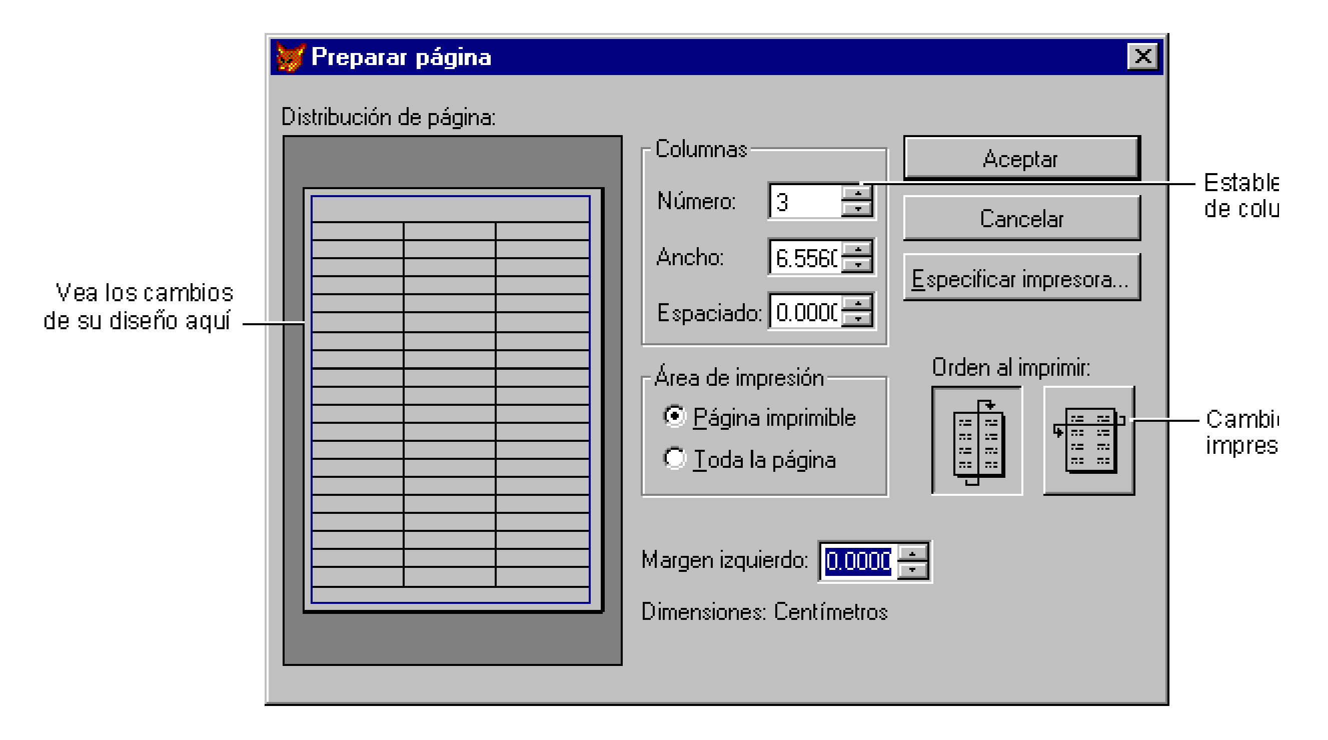
Definir múltiples columnas en una página

Para crear listas de teléfonos, etiquetas postales u otros tipos de listas, puede definir varias columnas por página.

Para definir un informe de múltiples columnas

1. En el menú **Archivo**, elija **Configurar página**.

   Cuadro de diálogo Configurar página con columnas definidas



2. En el área **Columnas**, escriba el número de columnas para la página. Es el mismo que el número de registros que desea imprimir en la página.

3. En el cuadro **Ancho**, escriba un valor para ancho de columna.

4. En el cuadro **Espacio**, escriba un valor para el espacio que desea que aparezca entre cada columna.

   Sugerencia Si imprime grupos que empiecen en una nueva página, no use la opción Orden al imprimir.

5. Elija **Aceptar**.

El **Diseñador de informes** refleja sus modificaciones.

Si el diseño ya contiene controles de informe en la banda Detalle, es posible que tenga que moverlos o cambiar su tamaño para que se ajusten a los límites de la nueva columna.

Establecer el alto de bandas de informe

Al diseñar el informe, puede cambiar el alto de una banda de informe. El alto de una banda de informe determina la cantidad de espacio que cada banda de informe usa en la página dentro de los márgenes de página. Por ejemplo, si la banda Título se establece a media pulgada (1,27 cm), el Título aparecerá en la primera media pulgada de espacio después del margen superior. La banda de detalle muestra la cantidad de espacio asignada para cada registro impreso. La siguiente información se aplica a todas las bandas del informe. Puede establecer parámetros adicionales para las bandas Encabezado de grupo y Pie. Para obtener información sobre las Bandas de grupo, consulte la sección "Agrupar datos en el diseño" en el capítulo 7, Diseño de informes y etiquetas, más adelante en este capítulo.

Para establecer un alto de banda exacta

1. Haga doble clic en la barra para la banda apropiada.

   Aparece un cuadro de diálogo para la banda.

2. En el cuadro **Alto**, escriba un valor para el alto.

3. Elija **Aceptar**.

Usar expresiones y funciones en controles de campo

Puede incluir controles de campo en su informe o etiqueta para mostrar valores de varias expresiones, incluyendo campos de tablas y vistas, variables y cálculos. Las siguientes secciones describen algunas expresiones y funciones usadas normalmente como campos múltiples, fechas o números de página.

Agregar controles de campo

Puede agregar controles de campo de varias maneras.

Para agregar campos de tabla del entorno de datos

1. Abra el entorno de datos del informe.

2. Seleccione una tabla o vista.

3. Arrastre campos al diseño.

Para agregar campos de la barra de herramientas

1. En la barra de herramientas **Controles de informes**, inserte un control **Campo**.

2. En el cuadro de diálogo Expresión de informe, elija el botón del cuadro de diálogo situado después del cuadro **Expresión**.

3. En el cuadro **Campos**, haga doble clic en el nombre del campo que desee.

   El nombre de tabla y el nombre de campo aparecen en el cuadro **Expresión para campo del informe**.

   Nota Si el cuadro **Campos** está vacío, agregue una tabla o vista al entorno de datos.

   No tiene que guardar el alias del nombre de tabla en la expresión. Puede eliminarlo o borrar las opciones del Generador de expresiones.

4. Elija **Aceptar**.

5. En el cuadro de diálogo **Expresión de informe**, elija **Aceptar**.

Después de escribir la expresión, puede cambiar el formato o establecer opciones de impresión, posición o ampliación. Para obtener más información, consulte "Agregar un comentario a un control" en el capítulo 7, Diseñar informes y etiquetas, del *Manual del usuario* y Establecer las opciones de impresión de un control más adelante en este mismo capítulo.

Insertar controles de campo concatenados

Después de agregar los campos de tabla, se dará cuenta de que no se imprimen de la forma deseada en la página. Por ejemplo, al imprimir los controles de campo para Ciudad, Región y Código postal de forma independiente aparecen espacios no deseados entre cada valor. Puede recortar o concatenar los campos de tabla en una expresión de campo. El espacio requerido por cada valor para este control variará. Puede establecer que el control se ajuste para cada valor.

Para combinar varios campos de tabla en una expresión

1. En la barra de herramientas **Controles de informes**, inserte un control **Campo**.

   Sugerencia Ajuste el tamaño del campo a la menor cantidad de espacio requerido por la expresión. Si se necesita más espacio, puede establecer el control para que se alargue para valores grandes, pero no puede reducirlo si necesita más espacio.

2. En el cuadro de diálogo Expresión de informe, seleccione el botón de diálogo situado junto al cuadro **Expresión**.

3. En el cuadro de diálogo Generador de expresiones, seleccione **ALLTRIM(expC)** en el cuadro **Cadena**.

   La función de cadena aparece en el cuadro **Expression** con `expC` seleccionado.

4. Haga doble clic en el nombre del primer campo que desea que aparezca en el control.

El nombre de campo reemplaza `expC`.

5. Escriba un signo más después del nombre de campo, seleccione un signo más después del nombre de campo o seleccione + en el cuadro de funciones **Cadena**.

6. Escriba o seleccione **Texto** en la lista Funciones de cadena y, a continuación, escriba una coma.

7. Repita los pasos 3 y 4 para agregar campos adicionales que completen la expresión y escriba **Aceptar**.

8. En el cuadro de diálogo **Expresión de informe**, seleccione **Ajustar al contenido del campo**.

   Cuando el control esté lleno, el espacio asignado al control se ajusta hacia abajo para alojar el valor de la expresión. Para obtener más información sobre **Ajustar al contenido del texto**, consulte Imprimir controles con valores de longitud variable, más adelante en este mismo capítulo.

Para combinar varios campos en una expresión, coloque una función ALLTRIM( ) antes de cada nombre de campo, coloque la puntuación entre comillas y coloque un signo menos entre cada elemento de la expresión. Si las longitudes de valores de campo no varían, como códigos postales o abreviaturas, puede insertar sólo el nombre de campo, como en este ejemplo:

```
ALLTRIM(ciudad)+", "+región+"  "+código_postal
```

Observe que se utilizan espacios entre comillas, en lugar de una coma, para separar la región y el código postal.

Para ver más ejemplos, vea el informe Invoice.frx en el directorio ...\Samples\Vfp98\Solution\Reports de Visual Studio.

Recortar y concatenar expresiones de caracteres

Para recortar y concatenar rápidamente expresiones de caracteres en el Generador de expresiones, puede colocar comas entre expresiones de caracteres. El valor de la expresión que precede a la coma se recorta. También puede usar punto y coma para colocar la expresión en una nueva línea, si el valor recortado tiene una longitud mayor que cero. El siguiente ejemplo muestra expresiones de caracteres para campos de una lista de distribución:

```
nombre_contacto; dirección; ciudad, región, código_postal
```

Nota   Úselos cuando no quiera incluir puntuación en el valor.

Si usa estos métodos, asegúrese de que el campo tiene establecido Ajustar al contenido del texto. Para obtener más información, consulte Imprimir controles con valores de longitud variable más adelante en este mismo capítulo.

Insertar la fecha actual

Puede insertar un control de campo que imprime la fecha actual.

Para insertar la fecha actual

1. En la barra de herramientas **Controles de informes**, inserte un control **Campo**.
2. En el cuadro de diálogo Expresión de informe, seleccione el botón de diálogo situado junto al cuadro **Expresión**.
3. En el Generador de expresiones, seleccione **DATE( )** en la lista **Fecha**.
4. Elija **Aceptar**.
5. En el cuadro de diálogo **Expresión de informe**, elija **Aceptar**.

Insertar un número de página

Las bandas Encabezado de página o Pie de página contienen normalmente un número de página. Si usa un asistente o Informe rápido, se inserta un número de página en la banda Pie de página.

Para insertar un número de página



1. En la barra de herramientas **Controles de informes**, inserte un control **Campo**.
2. En el cuadro Expresión de informe, seleccione el botón de diálogo situado junto al cuadro **Expresión**.
3. En el Generador de expresiones, seleccione **_pageno** en la lista **Variables**.
4. Elija **Aceptar**.
5. En el cuadro de diálogo **Expresión de informe**, elija **Aceptar**.

   Sugerencia Puede usar este procedimiento para insertar cualquiera de las variables de sistema de la lista Variables en el informe.

Definir variables de informe

Para manipular datos y mostrar el resultado de los cálculos en un informe, puede utilizar variables de informe. Puede calcular valores con variables de informe y usarlos después para calcular valores posteriores.

Para definir una variable de informe

1. Abre o crea un informe.

2. En el menú **Informe**, elija **Variables**.

3. En el cuadro de diálogo Variables del informe, seleccione el cuadro **Variables** y escriba un nombre para la variable.

4. En el cuadro **Almacenar valor**, escriba un nombre de campo o alguna otra expresión.

5. Si es necesario, seleccione una opción de cálculo.

6. Si es necesario, en el cuadro **Valor inicial**, escriba una expresión que establezca el valor inicial.

7. Elija **Aceptar**.

   Puede utilizar la variable en cualquier expresión que introduzca en el informe.

Para contar todas las entradas Canadá en la tabla Company, utilice esta expresión y seleccione "Recuento" como opción de cálculo.

```
IIF(country="Canadá",1,0)
```

El siguiente ejemplo muestra tres variables para una hoja de tiempos sencilla:

Para almacenar este valor	Cree esta variable	Mediante esta expresión
Hora de llegada del empleado	`tArrive`	`hour_in + (min_in / 60)`
Hora de salida del empleado	`tLeave`	`hour_out + (min_out / 60)`
Tiempo total que el empleado estuvo presente	`tDayTotal`	`tLeave - tArrive`

Puede utilizar la variable `tDayTotal` en una gran variedad de cálculos, tales como el número de horas trabajadas en una semana, un mes, o un año; el promedio de número de horas trabajadas cada día, etc.

Para ver ejemplos de variables de informe, vea los informes Percent.frx e Invoice.frx en el directorio ...\Samples\Vfp98\Solution\Reports de Visual Studio.

Reordenar variables de informe

Las variables de informe se evalúan en el orden en que aparecen en la lista y pueden afectar a los valores de las expresiones que las utilizan. Por ejemplo, si la variable 1 se utiliza para definir el valor de la variable 2, debe aparecer antes que la variable 2. En el ejemplo anterior de la hoja de tiempos, `tArrive` y `tLeave` deben preceder a `tDayTotal`.

Para cambiar el orden de las variables de informe

1. En el menú **Informe**, elija **Variables**.
2. En el cuadro **Variable**, arrastre el botón situado a la izquierda de la variable para cambiar el orden.
3. Elija **Aceptar**.

Establecer el valor inicial de una variable

Si utiliza una variable en cálculos, asegúrese de que inicializa dicha variable con un valor diferente de cero para no producir un error de división por cero. Si no especifica un valor, Visual FoxPro asignará un valor predeterminado.

Para establecer el valor inicial de una variable

1. En el menú **Informe**, elija **Variables**.
2. En el cuadro **Variable**, seleccione la variable que desea inicializar.
3. En el cuadro **Valor inicial**, introduzca el valor.
4. Elija **Aceptar**.

Si reordena los grupos en su informe, sus variables de informe podrían no restablecerse en el campo correcto. Por ejemplo, si su informe contiene dos grupos, el primero agrupado por país y el segundo agrupado por fecha, y cambia el orden de los grupos, las variables continuarán ajustadas de acuerdo a las posiciones originales de los grupos.

Puede cambiar el valor de un cálculo si especifica cuándo se restablece la variable. De forma predeterminada, Visual FoxPro restablece las variables de informe al final del informe.

Para restablecer una variable al final de un informe, una página o una columna

1. En el menú **Informe**, elija **Variables**.
2. En el cuadro **Restablecer**, elija una opción.
3. Elija **Aceptar**.

Para restablecer una variable al entrar o salir de alguna banda

1. En el Diseñador de informes, abra el informe.
2. Haga doble clic en la barra de la banda del informe que desee.
3. En el área **Ejecutar expresión** del cuadro de diálogo de la banda, elija uno de los botones al final del cuadro **Al entrar** o **Al salir**.

4. Escriba una expresión para restablecer la variable cada vez que se entre o se salga de esa banda.

Formato de controles de campo

Después de insertar un control de campo, puede cambiar el tipo de datos del control y el formato de impresión. Los tipos de datos pueden ser Character, Numeric o Date. Cada uno de estos tipos de datos tiene sus propias opciones de formato, incluyendo la opción de crear su propia plantilla de formato. El formato determina cómo se muestra el campo cuando se imprime el informe o la etiqueta.

Puede escribir funciones de formato directamente en el cuadro Expresiones del cuadro de diálogo Expresión de informe o seleccionar opciones en el cuadro de diálogo Formato.

Normalmente, puede convertir toda la salida alfabética a mayúsculas, insertar comas o comas decimales en salidas numéricas, mostrar salidas numéricas en formato de moneda o convertir un formato de fecha en otro.

Opciones de formato para controles de informe

Para controles de campo, puede establecer varias opciones de formato para cada tipo de datos.

Para dar formato a un control de campo

1. Elija el control **Campo**.

2. En el cuadro de diálogo Expresión de informe, elija el botón de diálogo situado junto al cuadro **Formato**.

3. En el cuadro de diálogo **Formato**, seleccione el tipo de datos para el campo: **Carácter**, **Numérico**, o **Fecha**.

   El área **Opciones de edición** muestra las opciones de formato disponibles para ese tipo de datos.

   Nota Este tipo de datos sólo se aplica al control del informe. Refleja el tipo de datos de la expresión y no cambia el tipo de datos del campo en la tabla.

4. Seleccione las opciones de alineación y formato que desee.

El cuadro de diálogo Formato muestra opciones diferentes en función del tipo de datos que elija. También puede crear una plantilla de formato si escribe caracteres en el cuadro Formato.

Alinear texto en un campo

Puede alinear el contenido de campos en controles de dos formas. Esta configuración no cambia la posición del control del informe, sólo el contenido en el espacio del control.

Para alinear el texto en un control de campo

1. Seleccione los controles que desea cambiar.

2. En el menú **Formato**, elija **Alineación de texto**.

3. En el submenú, elija el comando apropiado.

Para alinear texto en un campo

1. Elija el control **Campo**.

2. En el cuadro de diálogo Expresión de informe, elija el botón de diálogo situado junto al cuadro **Formato**.

Cuadro de diálogo Formato para una expresión de tipo Numérico



3. En el cuadro de diálogo **Formato**, seleccione el tipo de datos para el campo: **Carácter**, **Numérico** o **Fecha**.

4. Seleccione la alineación y las opciones de formato que quiera.

Definir plantillas de formato de campo

Una plantilla de formato le permite personalizar el formato del campo. Si escribe una combinación de caracteres y códigos en el cuadro Formato del cuadro de diálogo Expresión de informe o el cuadro de diálogo Formato, puede crear una amplia variedad de formatos de impresión. Los caracteres que escriba aparecen como texto junto con el valor del campo. Los códigos que escriba determinan la apariencia de la salida de campos. Por ejemplo, si usa la siguiente plantilla de formato para un campo numérico de 10 dígitos, los caracteres (paréntesis, espacios y guiones) se imprimen junto con los datos numéricos.

Plantilla de formato	Salida impresa
(999) 999-9999	(123) 456-7890

Cambio de fuentes

Puede cambiar el tamaño de fuente y el tamaño del texto para cada control campo o etiqueta, o puede cambiar la fuente predeterminada para el informe.

Para cambiar las fuentes y el tamaño en un informe

1. Seleccione el control.
2. En el menú **Formato**, seleccione **Fuente**.

   Aparece el cuadro de diálogo **Fuente**.

3. Seleccione la fuente apropiada y el tamaño en puntos y, a continuación, elija **Aceptar**.

Para cambiar la fuente predeterminada

1. En el menú **Informe**, elija **Fuente predeterminada**.
2. En el cuadro de diálogo **Fuente**, seleccione la fuente apropiada y el tamaño en puntos que desee como valores predeterminados y, a continuación, elija **Aceptar**.

   Sólo los controles insertados después de haber cambiado la fuente predeterminada reflejarán la nueva configuración de fuente. Para cambiar objetos existentes, selecciónelos a todos y cambie la fuente con la opción Fuente en el menú Formato.

Cortar una imagen u objeto OLE

Es posible que la imagen o el objeto OLE que haya insertado no se ajuste al marco dibujado al crear el control. De forma predeterminada, la imagen o el objeto conservan su tamaño original. Puede recortarlo o reducirlo para que se ajuste a su marco.

Si la imagen u objeto OLE es mayor que el marco creado en el Diseñador de informes, sólo aparecerá en el marco una porción de la imagen o el objeto. La imagen o el objeto están acoplados a la parte superior izquierda del marco. No puede ver la parte inferior derecha que se extiende más allá del marco.

Para ajustar una imagen al marco

1. En el Diseñador de informes, cree un Control Imagen/Control OLE dependiente.
2. En el cuadro de diálogo Imagen para informe, seleccione **Cambiar la escala de la imagen, conservar la forma**.

Aparece la imagen completa, que llenará la mayor parte del marco y conservará las posiciones relativas. Esto protege a la imagen de la distorsión vertical u horizontal.

Para rellenar el marco con la imagen

1. En el **Diseñador de informes**, cree un control **Imagen/Control OLE dependiente**.

2. En el cuadro de diálogo **Imagen para informe**, seleccione **Cambiar la escala de la imagen, rellenar el marco**.

La imagen cambia para llenar el marco que ha cambiado de tamaño. Si es necesario, la imagen se amplía verticalmente u horizontalmente para ajustarse al marco.

Para ver un ejemplo de un informe con imágenes, vea el informe Wrapping.frx en el directorio ...\Samples\Vfp98\Solution de Visual Studio.

Centrar un objeto OLE

Los objetos OLE incluidos en un campo General pueden variar en forma y tamaño. Si el objeto de un campo General es más pequeño que el marco, aparece en la esquina superior izquierda del marco. Puede centrarlo para asegurar que todos los objetos más pequeños que el marco estén centrados en el marco del informe o de la etiqueta. Las imágenes de archivo no están centradas porque no varían.

Para centrar objetos OLE de campo General

1. En el Diseñador de informes, cree un control Imagen/Control OLE dependiente.

2. En el cuadro de diálogo Imagen para informe, seleccione **Centrar imagen**.

Los objetos OLE impresos se centran en el área en una vista previa del informe o al imprimirlo.

Cambiar colores de controles de informe

Puede cambiar el color de un campo, etiqueta, línea o rectángulo.

Para cambiar los colores

1. Seleccione los controles que desea modificar.

2. En la barra de herramientas **Paleta de colores**, elija **Color de primer plano** o **Color de fondo**.

3. Seleccione el color que desee.

Guardar un informe como HTML

Puede usar la opción **Guardar como HTML** del menú **Archivo** cuando cree un informe para guardar el contenido de un formulario como un archivo HTML (Lenguaje de marcado de hipertexto).

Para guardar un informe como HTML

1. Abra el informe.

2. Elija **Guardar como HTML** en el menú **Archivo**. (Se le pedirá que guarde el informe si lo ha modificado).

3. Escriba el nombre del archivo HTML que va a crear y elija **Guardar**.

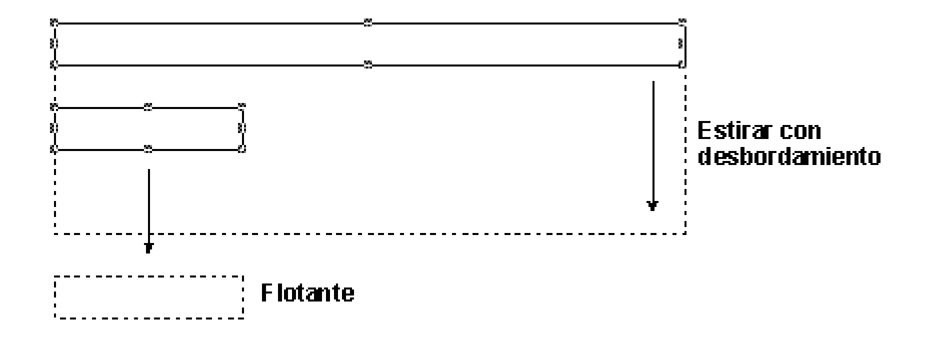
Opciones de impresión para controles

El diseño general y la posición de bandas de los controles determinan el momento y el lugar en que se imprimen. También puede establecer opciones de impresión específicas para cada control. Cada control puede tener un tamaño predeterminado en base a su valor (campos y tablas) o el tamaño creado (líneas, rectángulos e imágenes). La longitud del control en el diseño define el ancho de presentación del control. Como el valor de algunos valores varía de registro a registro, puede establecer el alto del control para alargar hacia abajo y mostrar así el valor completo. Si no establece que se amplíe, el valor se truncará en el ancho de presentación. No puede cambiar el tamaño de controles Etiqueta, pero puede cambiar el tamaño de los demás controles.

Imprimir controles con valores de longitud variable

Si quiere que un control sólo use el espacio necesario, puede establecer que se alargue. Por ejemplo, los valores de una expresión pueden variar de registro a registro. En lugar de asignar una cantidad fija de espacio en el informe que aloje el valor más largo, puede establecer que el control se alargue hacia abajo para mostrar todo el valor. Puede establecer que los controles situados por debajo del control que se amplía floten por debajo de la página con respecto al control que se ajusta.

Ejemplos de controles que se amplían y controles que flotan



La opción de alargar está disponible para campos, líneas verticales, rectángulos y rectángulos redondeados.

Para ver un ejemplo de controles que se amplían y flotan, vea el informe Wrapping.frx en el directorio ...\Samples\Vfp98\Solution de Visual Studio.

Para establecer que un campo se amplíe con su valor

1. Haga doble clic en el control campo para mostrar su cuadro de diálogo.

2. Seleccione **Ajustar al contenido del texto**.

Hay que establecer que los controles que se colocan con respecto a controles que se amplían floten, o se sobrescribirán.

Para establecer que un control flote

1. Haga doble clic en el control para mostrar su cuadro de diálogo.

2. En el cuadro de diálogo del control, seleccione **Flotante**.

   Precaución Algunos de los datos no se han podido sobrescribir durante la impresión si: (1) coloca un campo respecto al fondo de la banda e incluye *por debajo* de este campo otro campo que está colocado respecto a la parte superior de la banda y se puede alargar; o (2) coloca un campo respecto a la parte superior de la banda e incluye *por encima* de este campo otro campo colocado respecto a la parte superior de la banda y que se puede alargar.

También puede establecer que se alarguen líneas, rectángulos y rectángulos redondeados. Se pueden alargar con respecto a la banda o, si forman parte de un grupo de controles, se pueden alargar con respecto al control más grande del grupo.

Para establecer que se alargue una línea o un rectángulo

1. Haga doble clic en el control para mostrar su cuadro de diálogo.

2. En el área **Alargar hacia abajo**, seleccione una opción.

Para imprimir un borde en torno a un control que se puede alargar

1. Dibuje un rectángulo alrededor de los controles que se pueden alargar.

2. Haga doble clic en el rectángulo para mostrar el cuadro de diálogo Rectángulo/Línea.

3. En el área **Alargar hacia abajo**, seleccione **Alargar con relación al objeto más alto del grupo**.

4. Elija **Aceptar**.

5. Arrastre un cuadro de selección en torno al rectángulo.

6. En el menú **Formato**, elija **Agrupar**.

   Los controladores de selección aparecen en las esquinas del rectángulo. A partir de este momento puede tratar todos los controles como uno sólo. El rectángulo se alargará con el campo de tamaño ajustable. Independientemente de cuánto se alargue el valor del campo, el rectángulo mantendrá su borde alrededor del campo. Puede colocar dos de estos grupos lado a lado y uno no se verá afectado por el alargamiento del otro.

Para imprimir un control de tamaño ajustable por debajo de otro

1. Inserte los dos controles en el diseño, uno debajo del otro.

2. Haga doble clic en la parte superior del control para mostrar el cuadro de diálogo del control.

3. En el área **Posición del campo**, seleccione **Borde superior de la banda** y, a continuación, elija **Aceptar**.

4. Haga doble clic en la parte inferior del control para mostrar el cuadro de diálogo del control.

5. En el área **Posición del campo**, seleccione **Flotante** y, a continuación, elija **Aceptar**.

Los dos valores de registro se imprimirán completamente y no se sobrescribirán.

Establecer las opciones de impresión de un control

Puede controlar cuándo y cómo se imprime cada control de informe en el informe. Para obtener más información sobre las opciones de impresión, consulte cuadro de diálogo Imprimir-Condiciones.

Suprimir valores repetidos

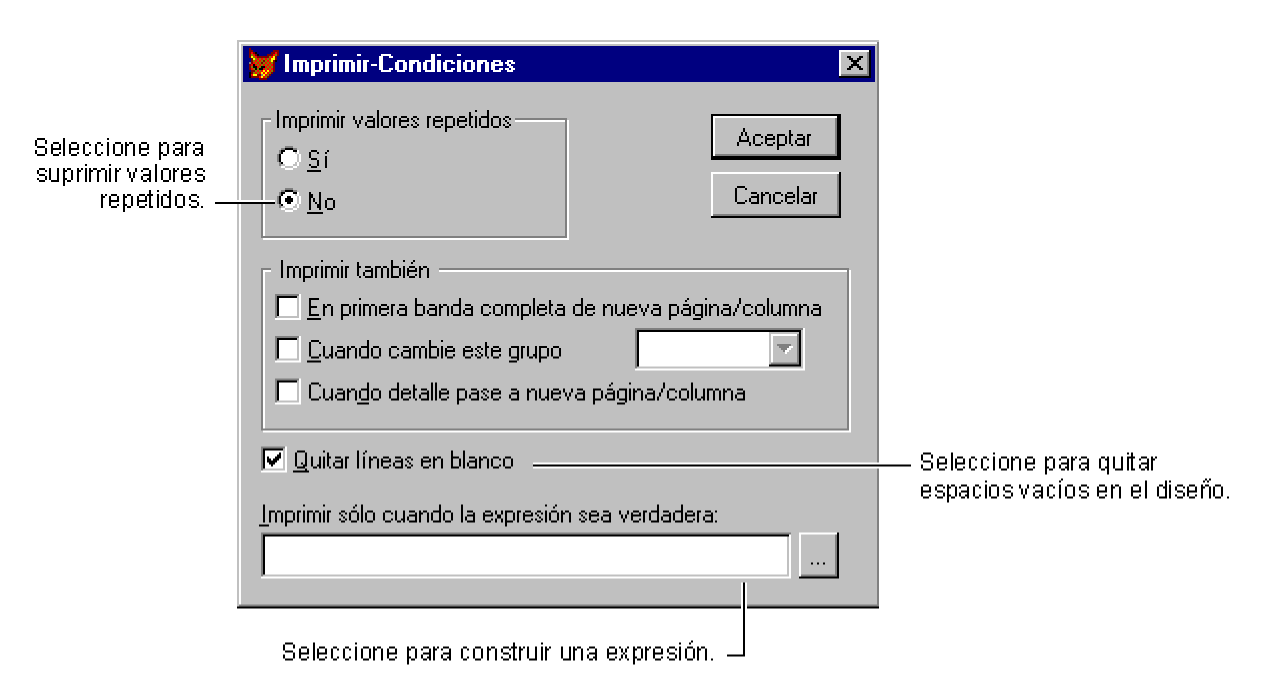
Para controles de campo, puede suprimir los valores repetidos para registros consecutivos de forma que el valor se imprima una vez para el primer registro pero no aparezca en los siguientes registros hasta que cambie. Por ejemplo, si va a imprimir una factura y uno de los campos contiene la fecha de la transacción, la fecha sólo se imprime una vez para transacciones que tuvieron lugar en esa fecha.

Para suprimir valores repetidos

1. Haga doble clic en el control para mostrar el cuadro de diálogo del control.

2. Elija **Imprimir-Condiciones** para mostrar el cuadro de diálogo **Imprimir-Condiciones**.

Cuadro de diálogo Imprimir-Condiciones



3. En el área **Imprimir valores repetidos**, seleccione **No** y, a continuación, elija **Aceptar**.

Para repetir en sólo una nueva página o columna

1. Haga doble clic en el control.
2. Elija **Imprimir-Condiciones**.
3. En el área **Imprimir valores repetidos**, seleccione **No**.
4. En el área I**mprimir también**, seleccione **En primera banda completa de nueva página/columna** y, a continuación, elija **Aceptar**.

Para repetir cuando se desborda una banda de detalle a una nueva página o columna

1. Haga doble clic en el control.
2. Elija **Imprimir-Condiciones**.
3. En el área **También imprimir**, seleccione **Cuando detalle pase a nueva página/columna** y, a continuación, elija **Aceptar**.

Generar expresiones de impresión

Puede agregar expresiones a un control: se evalúan antes de imprimir el campo. Si la expresión se evalúa a falso (.F.), el campo no se imprimirá. Si agrega una expresión, se desactivan todas las demás opciones del cuadro de diálogo Imprimir-Condiciones excepto Quitar líneas en blanco.

Para ver ejemplos de condiciones Imprimir-Condiciones, vea los informes Colors.frx y Ledger.frx del ejemplo Soluciones.

Para agregar una expresión de impresión

1. Haga doble clic en el control.
2. Elija **Imprimir-Condiciones**.
3. En el cuadro **Imprimir sólo cuando la expresión sea verdadera**, escriba una expresión.

   –O bien–

   Haga clic en el botón del diálogo para crear una expresión con el Generador de expresiones.

Suprimir líneas en blanco

El informe puede contener registros que no contengan valores para cada control de campo. De forma predeterminada, Visual FoxPro deja en blanco el área para ese campo. Puede quitar estas áreas en blanco para crear una presentación de la información más agradable y continua.

Para suprimir líneas en blanco

1. Haga doble clic en el control que produce las líneas en blanco en el informe.

2. Elija **Imprimir-Condiciones**.

3. Seleccione **Quitar líneas en blanco**.

Visual FoxPro quitará la línea del informe si se evalúa como línea en blanco. Si el campo no se imprime o si el campo de tabla está vacío, Visual FoxPro comprueba si hay otros controles en la línea. Si no se encuentra ninguno, se quita la línea. Si no selecciona esta opción y no hay otros controles en la misma línea, se imprime una línea en blanco.

Establecer opciones de impresión para grupos

Puede controlar cómo se imprimen grupos en el informe. A veces querrá que cada grupo empiece en una página distinta, o controlar cuándo se imprime el encabezado de grupo.

Establecer saltos de página de grupos

Además de seleccionar el campo o expresión que hay que agrupar, el cuadro de diálogo Agrupar datos le permite especificar opciones de salto de página para grupos.

Elegir una opción de encabezado de grupo

Puede que quiera que los grupos aparezcan en la siguiente columna para informes con varias columnas, en una nueva página para formularios o con una nueva página numerada como 1. El cuadro de diálogo Agrupar datos ofrece cuatro opciones para llevar a cabo estas tareas. Puede:

- Iniciar un grupo en una nueva columna.
- Iniciar cada grupo en una nueva página.
- Restablecer el número de página a 1 para cada grupo.
- Volver a imprimir el encabezado de grupo en cada página.

Después de escribir una expresión, puede seleccionar estas opciones en el área Propiedades de grupo.

Evitar encabezados de grupo huérfanos

A veces un grupo puede imprimirse parcialmente en una página y terminar en la siguiente. Para evitar que un encabezado de grupo se imprima cerca del final de la página con la mayor parte de los registros en la página siguiente, puede establecer la distancia mínima de la parte inferior a la que se va a imprimir un encabezado de grupo. Si el encabezado se coloca más cerca de la parte inferior de la página que el número de pulgadas o centímetros que haya escrito, Visual FoxPro imprime el encabezado en una nueva página.

Para evitar encabezados de grupo huérfanos

1. En el menú **Informe**, elija **Agrupar datos**.

2. En el cuadro de diálogo **Agrupar datos**, elija o escriba un valor en el cuadro **Comenzar cada**

grupo en una nueva página.

Sugerencia Para determinar un valor para un control huérfano, agregue el alto del Encabezado de grupo de una a tres veces al alto de Detalle.

Imprimir valores suprimidos cuando el grupo cambia

Si se suprimen valores repetidos, es posible que desee imprimir cuando cambie un grupo concreto.

Para imprimir valores repetidos cuando el grupo cambia

1. Haga doble clic en el control para mostrar el cuadro de diálogo del control.

2. Elija el botón **Imprimir-Condiciones** para imprimir el cuadro de diálogo **Imprimir-Condiciones**.

3. Seleccione **Cuando cambie este grupo**.

   Los grupos definidos para el informe aparecen en el cuadro.

4. Seleccione un grupo del cuadro y, a continuación, elija **Aceptar**.

Repetir encabezados de grupo

Cuando un grupo continúa en la página siguiente, es posible que quiera que el encabezado de grupo se repita en la parte superior del grupo para ver información continuada. Si tiene varios grupos de datos en el informe, el encabezado de las páginas subsiguientes será el del último grupo de la lista de grupos. Coloque todos los controles que desee imprimir para el encabezado de grupo en la banda de encabezado del último grupo de la lista.

Para repetir el encabezado de grupo en la página siguiente

- En el cuadro de diálogo Agrupar datos, seleccione el grupo que desea repetir y, a continuación, elija **Volver a imprimir el encabezado de grupo en cada página**.

  Si no desea repetir el encabezado de grupo, desactive esta casilla de verificación.

Controlar el resultado de informes y etiquetas

Puede controlar dónde se envía el resultado de un informe o una etiqueta si utiliza alguna de las siguientes palabras clave con el comando REPORT o LABEL:

- PRINT
- PREVIEW
- FILE

Si no utiliza una de estas palabra clave, el informe se enviará a la pantalla o a la ventana activa.

Seleccionar registros para imprimir

Cuando imprima un informe, es posible que desee limitar el número de registros que aparecen en el informe mediante criterios de selección. Puede:

- Elegir el alcance de registros si especifica una cantidad o intervalo.
- Generar una expresión FOR que seleccione registros que cumplan una condición.
- Generar una expresión WHILE que seleccione registros hasta que se encuentre uno que no cumpla una condición.

Puede usar cualquier combinación de estas opciones. La expresión WHILE pasa por alto los otros criterios.

Imprimir una cantidad o un intervalo de registros

Puede limitar el número de registros si especifica una cantidad o un intervalo de registros. Con la opción Alcance, puede seleccionar un único registro o un grupo de registros colocados secuencialmente en el archivo.

Nota El índice activo y el puntero de registro activo afecta a los resultados de las opciones de alcance Siguiente y Resto. Por ejemplo, el siguiente registro de una tabla indexada por el último nombre es probablemente diferente que el de una tabla indexada por estado. Esto no afecta a la opción Registro porque el número para un registro no cambia cuando se indexa la tabla.

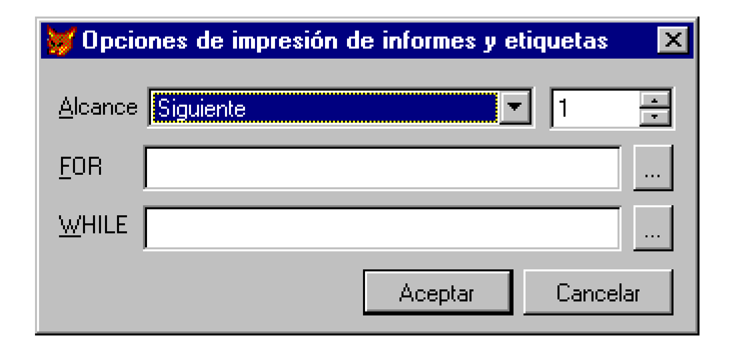
Para seleccionar un número limitado de registros

1. En el menú **Archivo**, elija **Imprimir**.

2. En el cuadro de diálogo **Imprimir**, elija **Opciones**.

3. En el cuadro de diálogo **Opciones de impresión**, elija **Opciones**.

   Cuadro de diálogo Opciones de impresión de informes y etiquetas

4. En el cuadro de diálogo **Opciones de impresión de informes y etiquetas**, elija **Alcance**.

5. Seleccione la opción de alcance apropiada.

Para imprimir	Elija esta opción de alcance
Cada registro del archivo de origen	TODO
Un intervalo de registro que empieza por 1	SIGUIENTE
Un registro específico por número	REGISTRO
El registro actual más todos los que le siguen hasta el final del archivo	RESTO

Visual FoxPro imprime el informe con datos de los registros que tengan el alcance que haya seleccionado.

Imprimir registros que cumplen una condición

Si los registros que desea seleccionar no están ordenados secuencialmente en la tabla, puede generar una expresión lógica que especifique criterios de selección que debe cumplir un registro para que se imprima. Por ejemplo, puede elegir imprimir todos los registros con un valor concreto en un campo.

Para introducir criterios para seleccionar registros

1. En el menú **Archivo**, elija **Imprimir**.
2. En el cuadro de diálogo **Imprimir**, elija **Opciones**.
3. En el cuadro de diálogo **Opciones de impresión**, elija **Opciones**.
4. En el cuadro de diálogo **Opciones de impresión de informes y etiquetas**, elija **Alcance**.
5. En el cuadro **FOR**, escriba una expresión FOR.

   –O bien–

   Asegúrese de que los orígenes de registros usados por el informe están abiertos y, a continuación, elija el botón **FOR** para usar el Generador de expresiones.

   Nota No tiene que incluir el comando FOR en la expresión. Por ejemplo, escriba **country = "Canadá"** para ver únicamente los datos canadienses.

   Visual FoxPro evalúa todos los registros e imprime el informe con los registros que cumplen la condición de la expresión.

Controlar la selección de registros para imprimir

Al imprimir, puede especificar una condición que se tiene que cumplir para seguir la evaluación y selección de registros. Esta condición se escribe en una expresión WHILE. Mientras la expresión WHILE sea verdadera, Visual FoxPro procesa el origen de datos. Después de buscar un registro que no cumpla la condición, Visual FoxPro termina el proceso de evaluación e imprime los registros

seleccionados. Esta opción le permite seleccionar registros en base a información externa a los valores contenidos en los campos.

Sugerencia Si usa una expresión WHILE en un archivo que no se ha indexado, el proceso de selección puede terminar antes de evaluar todos los registros apropiados. Antes de imprimir el informe, asegúrese de que la tabla de origen tiene el índice activo apropiado para la expresión WHILE que desea usar.

Para escribir criterios para terminar la selección de registros

1. En el menú **Archivo**, elija **Imprimir**.
2. En el cuadro de diálogo **Imprimir**, elija **Opciones**.
3. En el cuadro de diálogo **Opciones de impresión**, elija **Opciones**.
4. En el cuadro de diálogo **Opciones de impresión de informes y etiquetas**, elija **Alcance**.
5. En el cuadro de diálogo **WHILE**, escriba una expresión WHILE.

   –O bien–

   Elija el botón **WHILE** para usar el Generador de expresiones.

   Nota No tiene que incluir el comando WHILE en la instrucción. Por ejemplo, escriba **sales > 1000** para ver sólo las ventas por encima de 1000 pesetas.

   Visual FoxPro imprime el informe con los registros que evalúa mientras la expresión sea verdadera.

Imprimir informes y etiquetas

Si desea enviar el informe a la impresora, puede enviarlo directamente o mostrar el cuadro de diálogo Especificar impresora.

Para enviar un informe a la impresora

- Utilice la palabra clave TO PRINTER del comando REPORT o LABEL.

Por ejemplo, el código siguiente envía el informe `MiInforme` a la impresora predeterminada y hace que no se muestre en la pantalla:

```
REPORT FORM MIINFORME.FRX TO PRINTER NOCONSOLE
```

Para mostrar el cuadro de diálogo Especificar impresora antes de imprimir el informe

- Utilice las palabras clave TO PRINTER PROMPT del comando REPORT o LABEL.

Por ejemplo, el código siguiente muestra el cuadro de diálogo Especificar impresora, luego envía el

informe `MiInforme` a la impresora predeterminada y detiene la impresión del informe en la ventana activa:

```
REPORT FORM MIINFORME.FRX TO PRINTER PROMPT NOCONSOLE
```

Vista preliminar de informes y etiquetas

Si desea mostrar una vista preliminar del informe, puede enviarlo a la ventana Vista preliminar en el Diseñador de informes.

Para ver una vista preliminar de un informe

- Utilice la palabra clave PREVIEW del comando REPORT.

Por ejemplo, el código siguiente muestra el informe en una ventana modal:

```
REPORT FORM MIINFORME.FRX PREVIEW
```

De forma predeterminada, la ventana Vista preliminar es modal, pero permite tener acceso a la barra de herramientas. Si desea realizar la vista preliminar de forma no modal, puede agregar la palabra clave NOWAIT al comando REPORT.

Por ejemplo, el código siguiente muestra el informe en una ventana no modal:

```
REPORT FORM MIINFORME.FRX PREVIEW NOWAIT
```

Si desea ver los resultados en una ventana específica, puede incluir la cláusula WINDOW para especificar una ventana creada con DEFINE WINDOW.

```
REPORT FORM MIINFORME.FRX PREVIEW WINDOW MYWINDOW
```

Imprimir informes en un archivo

Si desea crear una versión electrónica del informe, puede enviarlo a un archivo con formato para su impresora o a un archivo ASCII. Si envía informes a archivos puede imprimirlos más tarde en un proceso por lotes.

Si desea crear un archivo ASCII, puede crear un archivo que incluya sólo el texto, guiones y signos más para representar líneas y formas. Los colores y las fuentes elegidas no están incluidos. También puede especificar el número de caracteres por línea y el numero de líneas por página.

Para imprimir un informe en un archivo ASCII

- Use las palabras clave FILE y ASCII del comando REPORT.

El ejemplo siguiente define las variables para la página ASCII e imprime un informe llamado Miinforme.frx en un archivo ASCII llamado. Miarch.txt.

Imprimir en un archivo ASCII

Programa	Comentario
`_asciirows = nLines`	Define el número de líneas por página.
`_asciicols = nChars`	Define el número de caracteres por línea.
`REPORT FORM MIINFORME.FRX` `TO FILEMIARCH.TXT ASCII`	Ejecuta el informe.

Guardar un informe como HTML

Puede usar la opción **Guardar como HTML** del menú **Archivo** cuando cree o modifique un informe para guardar su contenido como un archivo HTML (Lenguaje de marcado de hipertexto).

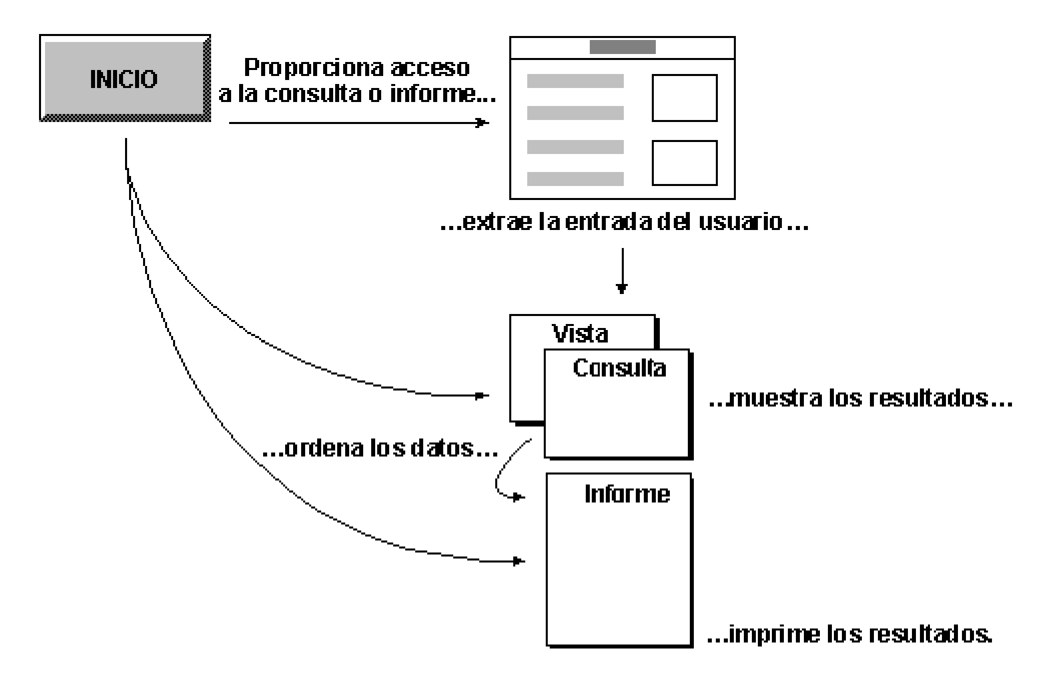
Para guardar un informe como HTML

1. Abra el informe.
2. Elija **Guardar como HTML** en el menú **Archivo**. Esta opción sólo está disponible cuando se ha guardado el informe en disco.
3. Escriba un nombre para el archivo HTML que desea crear y elija **Guardar**.

Integrar consultas e informes

Cuando haya creado los componentes de su aplicación, puede integrarlos. Esta ilustración muestra algunas formas en las que puede agregar consultas e informes a su aplicación.

Algunas formas de integrar consultas e informes



Puede agregar código que ejecute una consulta o un informe a los siguientes objetos de su aplicación.

- Un botón de un formulario. Para ver más información acerca de formularios y botones, consulte el capítulo 9,Crear formularios.
- Un elemento de un menú. Para ver más información acerca de cómo agregar elementos a un menú, consulte el capítulo 12, Diseñar menús y barras de herramientas.

Para agregar una consulta, vista o programa

- Agregue un comando DO o USE al código subyacente de un botón de comando de un formulario, un botón de una barra de herramientas o un elemento de menú.

  Por ejemplo, agregue código similar al de las siguientes líneas:

  ```
  DO MICONS.QPR
  DO MIPROG.PRG
  USE mivista
  ```

Dispone de varias opciones para integrar informes en su aplicación.

- Si desea que el usuario simplemente inicie el informe y lo recoja impreso, puede hacer que el usuario inicie el informe si agrega el comando REPORT a un control de un formulario, un comando en un menú o un botón de una barra de herramientas.
- Si desea permitir al usuario escribir algunas variables utilizadas en el informe, puede recoger los valores introducidos, de la misma manera que hizo antes para las consultas. Por ejemplo, un usuario podría introducir un intervalo de fechas que incluirá el informe. Para ver más información al respecto, consulte Recoger entradas de usuario con consultas más adelante en este mismo capítulo.
- Si desea que el usuario cree informes personalizados, puede ofrecerle la posibilidad de crear nuevos informes o modificar los ya existentes con el Diseñador de informes.

Para ejecutar informes y etiquetas

- Utilice los comandos REPORT o LABEL.

  Por ejemplo, podría utilizar código como el de las siguientes líneas:

  ```
  REPORT FORM MIINF.FRX
  LABEL FORM MIETIQL.LBX
  ```

Para modificar informes y etiquetas

- Utilice los comandos MODIFY REPORT o MODIFY LABEL.

  Por ejemplo, agregue código similar al de las siguientes líneas:

  ```
  MODIFY REPORT MIINF.FRX
  MODIFY LABEL MIETIQ.LBX
  ```

Para crear informes y etiquetas

- Utilice los comandos CREATE REPORT o CREATE LABEL.

  Por ejemplo, podría utilizar código similar al de las siguientes líneas:

  ```
  CREATE REPORT MIINF.FRX
  CREATE LABEL MIETIQ.LBX
  ```

Recoger entradas de usuario con consultas

Si quiere recoger valores desde un formulario, puede utilizar variables en una instrucción SELECT - SQL y luego utilizarlos en la instrucción, o ejecutar la instrucción más tarde.

Para recoger valores para uso inmediato, puede utilizar explícitamente el nombre del formulario o una referencia abreviada para el formulario en su instrucción SELECT - SQL. En este ejemplo, la referencia abreviada está en la cláusula WHERE.

Recogida de valores mediante referencias abreviadas en una instrucción SELECT - SQL

Código	Comentario

`SELECT * ;`
`   FROM tastrade!customer ;`
`   WHERE customer.country = ;`
`    THISFORM.ControlName1.Value ;`
`   AND customer.region =`
`THISFORM.ControlName2.Value ;`
`   GROUP BY customer.postal_code ;`
`   ORDER BY customer.postal_code,`
`customer.company_name`
	Utilice `THISFORM` como referencia abreviada para el formulario actualmente activo y sustituya los nombres de los controles por `ControlName1` y `ControlName2`.

Si no quiere utilizar referencias para los controles, puede definir variables en el código. Use variables en el código si desea almacenar los valores desde un formulario pero no utilizarlos necesariamente mientras el formulario esté activo.

Recogida de valores para uso posterior

Código	Comentario
`cValue = THISFORM.ControlName.Value`	Defina la variable.

`SELECT * ;`
`   FROM tastrade!customer ;`
`   WHERE customer.country = cValue ;`
`   GROUP BY customer.postal_code ;`
`   ORDER BY customer.postal_code, ;`
`     customer.company_name`
	Utilice la variable que ha definido en la instrucción SELECT - SQL.

Si no define la variable antes de ejecutar la consulta, aparecerá un mensaje de error que indique que la variable no se ha podido encontrar. Si la variable no está definida en el código, Visual FoxPro supondrá que la variable está pre-inicializada.

Capítulo 13: Compilar una aplicación

Puede crear fácilmente aplicaciones orientadas a objetos controladas por eventos, de una en una. Este enfoque modular le permite comprobar la funcionalidad de cada componente a medida que lo crea. Cuando haya creado todos los componentes funcionales, podrá compilarlos en una única aplicación, que ensamblará los componentes ejecutables del proyecto (formularios, informes, menús, programas, etc.) en un único archivo que podrá distribuir a los usuarios junto con los datos.

Para crear rápidamente un proyecto completo con el Marco de aplicaciones, puede usar el Asistente para aplicaciones. Una vez creado el proyecto, el Generador de aplicaciones abrirá el nuevo proyecto de modo que pueda agregar una base de datos, tablas, informes y formularios.

En este capítulo se describe la forma de generar una aplicación típica de Visual FoxPro. Para obtener más información sobre el proceso de programación de aplicaciones de Visual FoxPro, consulte el capítulo 2, Programar una aplicación, y el capítulo 14, Probar y depurar aplicaciones. Si desea distribuir la aplicación, vea la parte 8, Distribuir aplicaciones.

El proceso de generación de aplicaciones requiere lo siguiente:

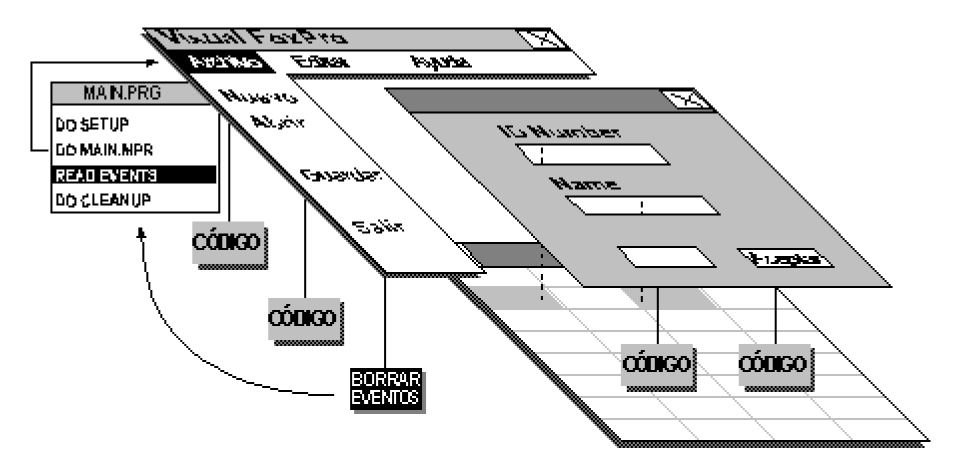
- Estructurar una aplicación
- Agregar archivos a un proyecto
- Generar una aplicación a partir de un proyecto

Estructurar una aplicación

Una aplicación típica de base de datos consta de estructuras de datos, una interfaz de usuario, opciones de consulta y funciones de generación de informes. Para diseñar la estructura de su aplicación, considere detenidamente la función que ofrece cada componente y su relación con los demás componentes.

Una aplicación ensamblada de Visual FoxPro suele presentar al usuario un menú y uno o más formularios para introducir o mostrar datos. Para ofrecer determinada funcionalidad y mantener la integridad y seguridad de los datos, adjunte código a determinados eventos. Las consultas y los informes permiten que los usuarios extraigan información de la base de datos.

Estructura de una aplicación típica de Visual FoxPro



A la hora de estructurar la aplicación, es necesario tener en cuenta las tareas siguientes:

- Establecer el punto de partida.
- Inicializar el entorno.
- Presentar la interfaz.
- Controlar el bucle de eventos.
- Restaurar el entorno original al terminar la aplicación.

En las secciones siguientes se proporcionan detalles sobre todas estas tareas. Normalmente podría crear un objeto Application para realizarlas; consulte la aplicación de ejemplo Importadores Tasmanian ubicada en el directorio ...\Samples\Vfp98\Tastrade de Visual Studio para ver un ejemplo de esta técnica. Asimismo, si usa el Asistente para aplicaciones para compilar la aplicación, se creará un objeto Application. También puede, si lo desea, usar un programa como archivo principal que controle estas tareas. Para obtener más información, vea "Estructurar un programa principal".

Establecer el punto de partida

Debe vincular todos los componentes y establecer un punto inicial para la aplicación con un archivo principal. El archivo principal sirve como punto de partida para la ejecución de su aplicación y puede constar de un programa o formulario. Cuando los usuarios ejecuten la aplicación, Visual FoxPro inicia el archivo principal de la aplicación que, a su vez, ejecuta todos los demás componentes a medida que se vayan necesitando. Todas las aplicaciones deben tener un archivo principal. La mejor elección suele ser crear un programa principal en la aplicación. Sin embargo, puede combinar las

funciones del programa principal y la interfaz inicial de usuario si usa un formulario como programa principal.

Si usa un Asistente para aplicaciones para crear la aplicación, puede permitir que el asistente cree un programa de archivo principal de forma automática. No es necesario que especifique un archivo principal a menos que desee cambiarlo una vez finalizadas las acciones del asistente.

Para establecer el punto de partida para una aplicación

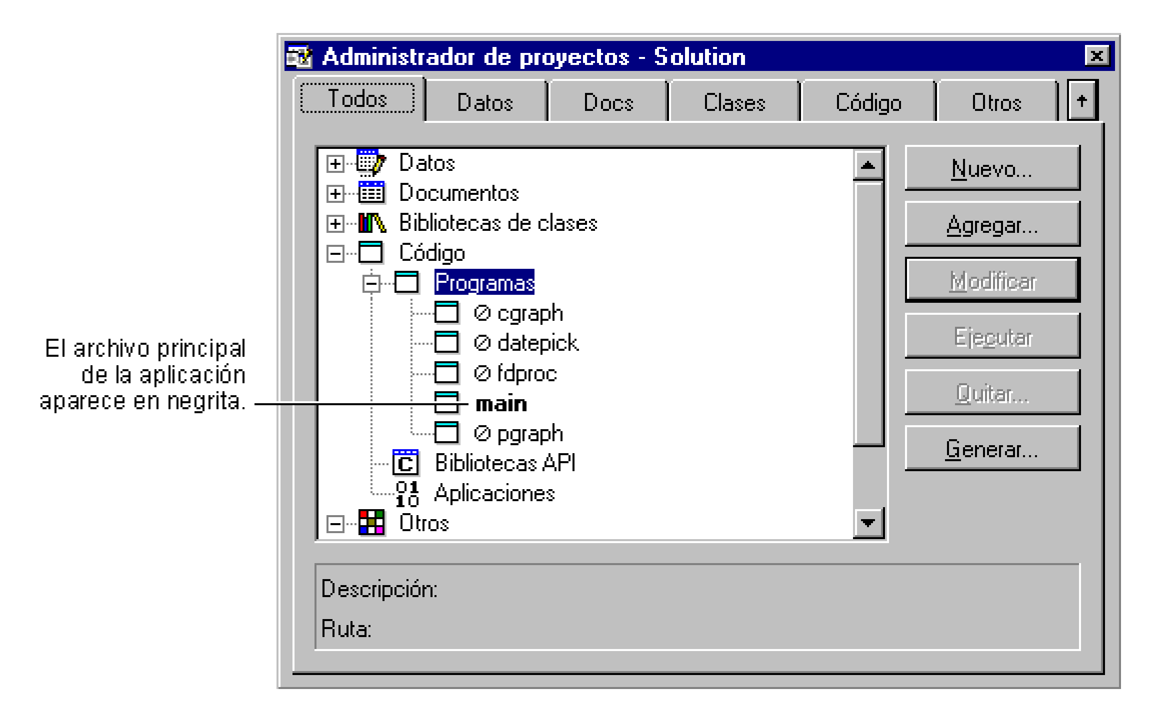
1. En el Administrador de proyectos, seleccione el archivo.

2. En el menú **Proyecto**, elija **Establecer principal**.

   Nota El archivo que establezca como archivo principal de la aplicación se marca automáticamente como incluido, por lo que se considera como de sólo lectura después de compilar la aplicación.

Sólo se puede establecer como archivo principal un archivo del proyecto. El archivo principal se muestra en negrita, como se muestra en la ilustración siguiente.

Establecer un archivo principal en el Administrador de proyectos



Inicializar el entorno

Cuando haya creado un archivo principal, utilícelo para configurar el entorno de su aplicación. Puede que el entorno de desarrollo predeterminado de Visual FoxPro no sea el entorno más adecuado para la aplicación. El entorno predeterminado establece determinados valores para los comandos SET y las variables de sistema en el momento de abrir Visual FoxPro.

Sugerencia Para ver los valores predeterminados del entorno de desarrollo de Visual FoxPro, inicie

Visual FoxPro sin ningún archivo de configuración; para ello, escriba **VFP6 -C** y, a continuación, ejecute el comando DISPLAY STATUS.

Siempre es conveniente guardar los valores iniciales del entorno y configurar un entorno específico para la aplicación en el preprograma.

Para capturar comandos para el entorno actual

1. En el menú **Herramientas**, elija **Opciones**.

2. Presione **Mayúsculas** y elija **Aceptar** para mostrar los comandos SET del entorno en la ventana **Comandos**.

3. En la ventana **Comandos**, copie y pegue en el programa.

En un entorno específico de la aplicación, podría incluir código para:

- Inicializar variables.
- Establecer una ruta predeterminada.
- Abrir las bases de datos, las tablas libres y los índices necesarios. Si la aplicación requiere acceso a datos remotos, también se puede solicitar al usuario la información de inicio de sesión necesaria en la rutina de inicialización.
- Hacer referencia a archivos externos de biblioteca y procedimientos.

Por ejemplo, si desea comprobar el valor predeterminado del comando SET TALK, almacenar este valor y establecer SET TALK como OFF para la aplicación, podría incluir el siguiente código en el procedimiento de configuración:

```
IF SET('TALK') = "ON"
   SET TALK OFF
   cTalkVal = "ON"
ELSE
   cTalkVal = "OFF"
ENDIF
```

Suele ser útil guardar los valores predeterminados en variables públicas, en una clase personalizada o como propiedades de un objeto Application para que pueda restaurar estos valores al salir de la aplicación.

```
SET TALK &cTalkVal
```

Mostrar la interfaz inicial

La interfaz inicial de usuario puede ser un menú, un formulario o cualquier otro componente de usuario. Normalmente una aplicación mostrará una pantalla de inicio de sesión o un cuadro de diálogo de inicio antes de mostrar el menú o formulario inicial.

Para iniciar la interfaz de usuario en el programa principal puede usar un comando DO para ejecutar un menú o un comando DO FORM para ejecutar un formulario.

Controlar el bucle de eventos

Cuando el entorno esté configurado y se muestre la interfaz de usuario inicial, podrá establecer un bucle de eventos para esperar la interacción del usuario.

Para controlar el bucle de eventos

- Ejecute un comando READ EVENTS, que hace que Visual FoxPro comience a procesar los eventos de usuario, como los clics del *mouse* y las pulsaciones de teclas.

Es importante situar correctamente el comando READ EVENTS en el archivo principal, porque todo el proceso de este archivo se suspende desde el momento en que se ejecuta el comando READ EVENTS hasta que se ejecuta un comando CLEAR EVENTS. Por ejemplo, podría ejecutar un comando READ EVENTS como el último comando de un procedimiento de inicialización, que se ejecutaría después de inicializar el entorno y mostrar la interfaz de usuario. Si no incluye el comando READ EVENTS, la aplicación volverá al sistema operativo después de la ejecución.

Una vez iniciado el bucle de eventos, la aplicación está bajo el control del último elemento mostrado de la interfaz de usuario. Por ejemplo, si se ejecutan los dos comandos siguientes en el archivo principal, la aplicación muestra el formulario Startup.scx:

```
DO FORM STARTUP.SCX
READ EVENTS
```

Si no incluye en el archivo principal un comando READ EVENTS o su equivalente, la aplicación se ejecutará correctamente desde la ventana Comandos dentro del entorno de programación. Sin embargo, cuando se ejecuta desde el menú o la pantalla principal, la aplicación aparecerá durante unos instantes y después finalizará.

La aplicación también debe proporcionar una forma de terminar el bucle de eventos.

Para terminar el bucle de eventos

- Ejecute un comando CLEAR EVENTS.

Normalmente el comando CLEAR EVENTS se ejecuta desde un menú o botón de un formulario. El comando CLEAR EVENTS suspende el proceso del evento en Visual FoxPro y devuelve el control al programa que ejecutó el comando READ EVENTS e inició el bucle de eventos.

Para ver un sencillo ejemplo de programa, vea Estructurar un programa principal más adelante en este mismo capítulo.

Precaución Necesita establecer una forma de salir del bucle de eventos antes de iniciarlo. Asegúrese de que la interfaz tiene un mecanismo (como un botón "Salir" o un comando de menú Salir) para ejecutar el comando CLEAR EVENTS.

Restaurar el entorno original

Para restaurar el valor original de las variables guardadas, puede sustituirlas mediante una macro en los comandos SET originales. Por ejemplo, si guardó la configuración de SET TALK en `cTalkVal`, ejecute el comando siguiente:

```
SET TALK &cTalkval
```

Nota Los nombres de variables utilizadas en la sustitución de macros no deben contener el prefijo "m." porque el punto presupone una concatenación de variables y producirá un error sintáctico.

Si inicializó el entorno en un programa que no sea aquél en el que se está realizando la restauración (por ejemplo, si inicializa llamando a un procedimiento, pero restaura llamando a otro) asegúrese de que puede tener acceso a los valores almacenados. Por ejemplo, almacene los valores para restaurarlos en variables públicas, clases personalizadas o como propiedades de un objeto Application.

Estructurar un programa principal

Si usa un archivo de programa (.prg) como archivo principal en la aplicación, debe comprobar que incluya los comandos que gestionarán las tareas asociadas con las principales tareas de la aplicación. El archivo principal no tiene que ejecutar comandos directamente para realizar todas las tareas. Por ejemplo, es frecuente llamar a procedimientos o funciones para gestionar tareas como inicialización del entorno y limpieza.

Nota Si usó el Asistente para aplicaciones y le permitió crear el programa Main.prg, puede modificar el programa creado por el asistente en lugar de crear otro nuevo. Los asistentes utilizan una clase especial para definir un objeto para la aplicación. El programa principal incluye secciones para crear instancias y configurar el objeto.

Para generar un único programa principal

1. Inicialice el entorno; para ello, abra bases de datos, declare variables, etc.
2. Establezca la interfaz inicial de usuario; para ello, llame a un menú o formulario.
3. Establezca el bucle de eventos; para ello, ejecute el comando READ EVENTS.
4. Ejecute el comando CLEAR EVENTS desde un menú (como un comando Salir) o un botón (como un botón del comando Salir). El programa principal no debería ejecutar este comando.
5. Restablezca el entorno cuando salga de la aplicación.

Por ejemplo, su programa principal podría ser similar al siguiente:

Código	Comentarios
`DO SETUP.PRG`	Llama al programa para configurar el entorno (almacena los valores en variables públicas)
`DO MAINMENU.MPR`	Muestra la interfaz inicial de usuario
`READ EVENTS`	Establece el bucle de eventos. Un programa diferente (como Mainmenu.mpr) debe ejecutar el comando CLEAR EVENTS)
`DO CLEANUP.PRG`	Restaura el entorno antes de salir

Agregar archivos a un proyecto

Un proyecto de Visual FoxPro consta de componentes independientes que se almacenan como archivos individuales. Por ejemplo, un proyecto sencillo puede constar de formularios (archivos .scx), informes (archivos .frx) y programas (archivos .prg y .fxp). Además un proyecto suele tener una o varias bases de datos (archivos .dbc), tablas (almacenadas en archivos .dbf y .fpt) e índices (archivos .cdx e .idx). Para incluirse en una aplicación, el archivo ha de agregarse al proyecto. De esa manera, al compilar la aplicación, Visual FoxPro puede incluir los archivos de ese componente en el producto terminado.

Puede agregar fácilmente archivos a un proyecto de varias maneras:

- Para crear un proyecto y agregar archivos existentes, use el Asistente para aplicaciones.
- Para agregar automáticamente archivos nuevos a un proyecto, abra un proyecto y, a continuación, cree los archivos nuevos en el Administrador de proyectos.
- Para agregar archivos existentes a un proyecto, abra un proyecto y agréguelos con el Administrador de proyectos.

Si ha utilizado el Asistente para aplicaciones o el Administrador de proyectos para crear los archivos, normalmente no necesitará hacer nada más ya que el archivo se incluye automáticamente en el proyecto. Sin embargo, existe una excepción a lo anterior si la aplicación incluye un archivo que el usuario vaya a modificar. Dado que los archivos incluidos son de sólo lectura, deberá marcar el archivo como excluido. Para obtener más detalles, vea el apartado Referencias a archivos modificables más adelante en este capítulo.

Sugerencia Para obtener una lista de los tipos de archivos y extensiones utilizados en Visual FoxPro, vea Extensiones y tipos de archivos.

Si un archivo existente ya no forma parte del proyecto, puede agregarlo manualmente.

Para agregar un archivo a un proyecto manualmente

1. En el Administrador de proyectos, elija el tipo de componente que desea agregar; para ello,

seleccíónelo en la jerarquía y, a continuación, haga clic en **Agregar**.

2. En el cuadro de diálogo **Abrir**, seleccione el archivo que desea agregar.

Visual FoxPro también agrega archivos al proyecto si se ha hecho referencia a los mismos en un programa o formulario. Por ejemplo, si un programa del proyecto incluye la línea siguiente, Visual FoxPro agrega el archivo Orders.scx al proyecto:

```
DO FORM ORDERS.SCX
```

Si se ha hecho referencia a un archivo de esta manera, no se incluye inmediatamente en un proyecto. Posteriormente, cuando compile el proyecto, Visual FoxPro resuelve las referencias a todos los archivos e incluye automáticamente los archivos implícitos en el proyecto. Además, si se ha hecho referencia a cualquier otro archivo mediante código definido por el usuario en el archivo nuevo, al compilar el proyecto también se resolverá esa referencia y se incluirá el archivo. Los archivos a los que se ha hecho referencia aparecerán en el Administrador de programas la próxima vez que vea el proyecto.

Importante Visual FoxPro podría no ser capaz de resolver referencias a archivos de imagen (.bmp y .msk), dependiendo de la forma en que se utilicen en el código. Por tanto, agregue las imágenes a los archivos manualmente. Además, Visual FoxPro no puede incluir automáticamente archivos a los que se haya hecho referencia mediante la sustitución de macros, porque el nombre del archivo no se conoce hasta que se ejecuta la aplicación. Si la aplicación establece referencias a archivos mediante la sustitución de macros, incluya estos archivos manualmente.

Referencias a archivos modificables

Cuando compile un proyecto en una aplicación, los archivos incluidos en el proyecto se ensamblan en un único archivo de aplicación. Una vez compilado el proyecto, los archivos del proyecto que estén marcados como "incluidos" pasarán a ser de sólo lectura.

Los archivos que forman parte del proyecto, como las tablas, suelen estar diseñados para que los usuarios puedan modificarlos. En esos casos, debería agregar los archivos al proyecto pero marcándolos como excluidos. Los archivos excluidos siguen formando parte de la aplicación por lo que Visual FoxPro los registra como parte del proyecto, pero no se compilan en el archivo de aplicación y, de este modo, los usuarios podrán actualizarlos.

Nota Como opción predeterminada las tablas se marcan como excluidas porque Visual FoxPro supone que las tablas serán modificables en una aplicación.

Como regla general, los archivos que contengan programas ejecutables (formularios, informes, consultas, menús y programas) deberán incluirse en el archivo de aplicación y los archivos de datos deberán excluirse. Sin embargo, ha de determinar si va a incluir o excluir archivos según los requisitos de la aplicación. Por ejemplo, una tabla que contenga información sobre el sistema o datos utilizados únicamente para consulta puede incluirse en el archivo de aplicación para proteger dicha información ante cambios que se realicen sin previo aviso. A la inversa, podría excluir un archivo de informe (.frx) si la aplicación permite a los usuarios modificarlo.

Si excluye un archivo, deberá asegurarse de que Visual FoxPro pueda encontrar el archivo excluido

cuando se ejecute la aplicación. Por ejemplo, cuando en un formulario se hace referencia a una biblioteca de clases visuales, el formulario almacena una ruta de acceso relativa a esa biblioteca. Si incluye la biblioteca en el proyecto, entrará a formar parte del archivo de aplicación y el formulario siempre podrá localizarla. Sin embargo, si excluye la biblioteca, el formulario debe buscarla usando la ruta de acceso relativa o la ruta de acceso de búsqueda de Visual FoxPro (definida mediante el comando SET PATH). Si la biblioteca no se encuentra en las ubicaciones previstas (por ejemplo, si se ha cambiado de ubicación desde que se creó el formulario) Visual FoxPro muestra un cuadro de diálogo en el que se solicita al usuario que localice la biblioteca. Tal vez desee que los usuarios no vean este cuadro de diálogo. Para estar seguro, incluya todos los archivos que no necesiten actualización por parte de los usuarios.

Nota No puede incluir archivos de aplicación (.app) y debe optar por excluir los archivos de biblioteca (.ocx, .fll y .dll).

Para excluir archivos modificables

1. Seleccione el archivo modificable en el Administrador de proyectos.

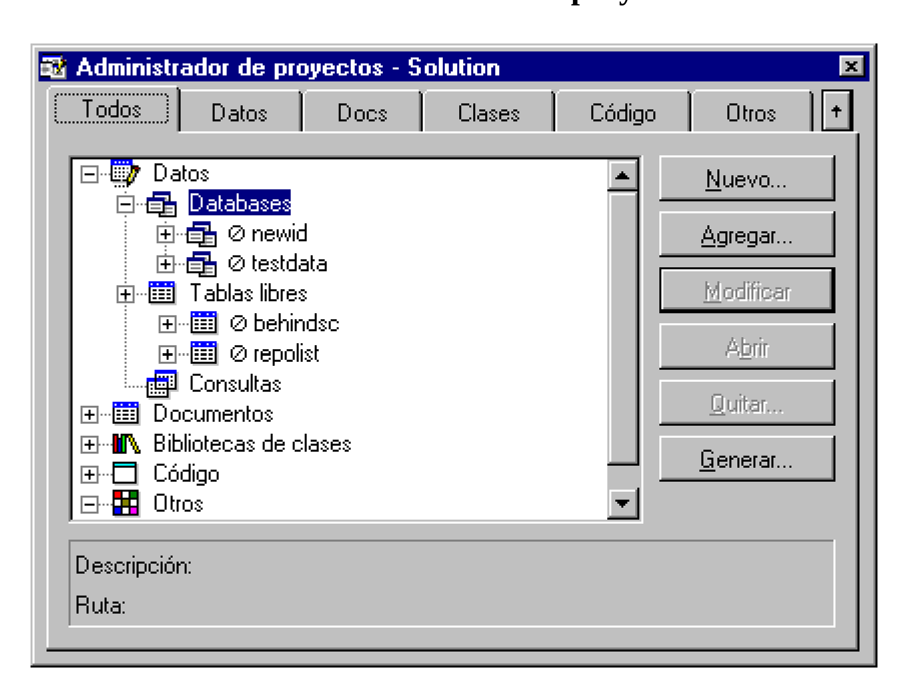
2. En el menú **Proyecto**, elija **Excluir**.

   Si el archivo ya está excluido, el comando **Excluir** no está disponible; aparecerá el comando **Incluir** en su lugar.

Los archivos excluidos tienen el símbolo ∅ a la izquierda de sus nombres.

Nota Los archivos marcados como archivos principales no se pueden marcar como excluidos. Para obtener detalles sobre los archivos principales, vea el apartado Establecer el punto de partida de este capítulo.

Tablas marcadas como excluidas en un proyecto



Sugerencia Para ver simultáneamente todos los archivos de proyecto, elija **Información de proyecto** en el menú **Proyecto** y seleccione la ficha **Archivos**.

Generar una aplicación a partir de un proyecto

Si puede generar el proyecto sin errores, podrá generar la aplicación. El resultado final de este proceso es un único archivo que incluye todos los archivos a los que se ha hecho referencia en el proyecto en un único archivo de aplicación (salvo los marcados como excluidos). Puede distribuir el archivo de aplicación, junto con los archivos de datos y con todos los demás archivos excluidos del proyecto a los usuarios y éstos pueden iniciar el archivo para ejecutar la aplicación.

Los pasos necesarios para crear una aplicación desde el proyecto son:

- Probar el proyecto.
- Generar un archivo de aplicación a partir del proyecto.

Probar el proyecto

Para comprobar las referencias y que todos los componentes están disponibles, puede probar el proyecto. Para ello, deberá volver a generar el proyecto. Visual FoxPro tendrá que resolver las referencias a archivos y volver a compilar los archivos no actualizados.

Para probar un proyecto

1. En el Administrador de proyectos, elija **Generar**.

2. En el cuadro de diálogo Opciones para generar elija **Volver a generar el proyecto**.

3. Seleccione cualquier otra opción que necesite y elija **Aceptar**.

   –O bien–

- Utilice el comando BUILD PROJECT.

Por ejemplo, para generar una aplicación denominada Miproy.pjx, escriba:

```
BUILD PROJECT miproy
```

Si se producen errores durante el proceso de generación, se incluyen en un archivo que recibe el nombre del proyecto y la extensión .err y se localiza en el directorio actual. El recuento de errores de compilación se muestra en la barra de estado. También puede ver el archivo de errores.

Para mostrar el archivo de errores

- Seleccione la casilla de verificación **Mostrar errores**.

Una vez generado el proyecto correctamente, deberá intentar ejecutarlo antes de crear una aplicación.

Para ejecutar una aplicación

- En el **Administrador de proyectos**, resalte el programa principal y, a continuación, elija **Ejecutar**.

  -O bien–

- En la ventana **Comandos**, ejecute un comando DO con el nombre del programa principal:

  ```
  DO MAINPROG.PRG
  ```

Si el programa se ejecuta correctamente, estará preparado para generar un archivo de aplicación que contenga todos los archivos incluidos en el proyecto.

Debería repetir los pasos de regeneración y ejecución del proyecto siempre que agregue componentes al proyecto. A menos que elija Volver a compilar todos los archivos en el cuadro de diálogo Opciones para generar, sólo se volverán a compilar los archivos que se hayan modificado desde la última generación.

Generar una aplicación desde un proyecto

Para generar un archivo terminado desde la aplicación, genérelo en un archivo de aplicación. Los archivos de aplicación tienen la extensión .app. Para ejecutar la aplicación, los usuarios deben iniciar Visual FoxPro primero y después cargar el archivo .app.

Puede optar por generar un archivo de aplicación (.app) o un archivo ejecutable (.exe) desde el proyecto. Los usuarios pueden ejecutar un archivo .app si ya tienen una copia de Visual FoxPro. Además, puede crear un archivo .exe, que funciona con dos bibliotecas de vínculos dinámicos de Visual FoxPro (Vfp6r.dll y Vfp6enu.dll), las cuales se distribuyen con la aplicación para ofrecer un entorno de ejecución completo para Visual FoxPro. El segundo archivo es específico del país de destino de la aplicación. Para obtener más información, vea la parte 8, Distribuir aplicaciones.

Para generar una aplicación

1. En el **Administrador de proyectos**, elija **Generar**.

2. En el cuadro de diálogo Opciones para generar, elija **Generar aplicación** si desea generar un archivo .app o **Generar ejecutable** si desea generar un archivo .exe.

3. Seleccione cualquier otra opción que necesite y elija **Aceptar**.

   –O bien–

- Utilice los comandos BUILD APP o BUILD EXE.

Por ejemplo, para generar una aplicación denominada Miapli.app a partir de un proyecto denominado Miproy.pjx, escriba:

```
BUILD APP miapli FROM miproy
```

Para crear una aplicación denominada Miapli.exe a partir de un proyecto llamado Miproy.pjx, escriba:

```
BUILD EXE miapli FROM miproy
```

Nota También puede usar el cuadro de diálogo Opciones de generación para crear un servidor de Automatización personalizado desde la aplicación de Visual FoxPro. Para obtener más información, vea Crear servidores de Automatización en el capítulo 16, "Agregar OLE".

Una vez creado y terminado un archivo de aplicación para el proyecto, todos los usuarios podrán ejecutarlo.

Para ejecutar una aplicación como un archivo .app

- En Visual FoxPro, elija **Ejecutar** en el menú **Programa** y seleccione el archivo de aplicación.

  –O bien–

- En la ventana **Comandos**, escriba DO y el nombre del archivo de aplicación.

  Por ejemplo, para ejecutar una aplicación denominada MIAPLI, escriba:

  ```
  DO miapli.app
  ```

Si ha creado un archivo .exe desde la aplicación, los usuarios podrán ejecutarlo de diversas maneras.

Para ejecutar una aplicación como un archivo .exe

- En Visual FoxPro, elija **Ejecutar** en el menú **Programa** y seleccione el archivo de aplicación o, en la ventana **Comandos**, escriba DO y el nombre del archivo de aplicación.

  Por ejemplo, para ejecutar un archivo .exe denominada Miapli.exe, escriba:

  ```
  DO miapli.exe
  ```

  –O bien–

- En Windows, haga doble clic en el icono del archivo .exe.

  Nota Puede usar el Asistente para instalación para crear una rutina de instalación que instale los archivos apropiados.

Capítulo 14: Probar y depurar aplicaciones

El proceso de prueba implica la búsqueda de problemas en el código, mientras que la depuración

consiste en aislar y resolver los problemas. La prueba y la depuración son partes inevitables del ciclo de desarrollo, que es mejor incorporar en una etapa inicial. La prueba y depuración de componentes individuales simplifica de forma significativa el proceso análogo de aplicaciones integradas.

Para obtener más información sobre la creación de una aplicación, consulte el capítulo 2, Programar una aplicación y el capítulo 13, Compilar una aplicación.

Este capítulo trata los temas siguientes:

- Preparar la prueba y depuración
- Depurar antes de que se produzcan errores
- Aislar los problemas
- Mostrar los resultados
- Registrar el trayecto del código
- Controlar errores de tiempo de ejecución

Preparar la prueba y depuración

Los programadores suelen buscar diferentes niveles de solidez cuando comprueban y depuran sus aplicaciones:

1. Ejecución sin bloqueos ni generación de mensajes de error.

2. Acciones apropiadas en situaciones frecuentes.

3. Acciones razonables o mensajes de error en diversas situaciones.

4. Recuperación inmediata ante interacciones imprevistas del usuario.

Visual FoxPro ofrece un variado conjunto de herramientas para facilitar la tarea de aislar e identificar los problemas del código para que puedan resolverse de forma eficaz. Sin embargo, una de las mejores maneras de crear aplicaciones robustas es buscar los posibles problemas antes de que se produzcan.

Depurar antes de que se produzcan errores

La teoría ha demostrado que las buenas costumbres de escritura de código (uso de espacios en blanco, inclusión de comentarios, adhesión a las convenciones de nombres, etc.) tienden a reducir automáticamente el número de errores del código. Además, hay algunos pasos que se pueden realizar en el proceso de programación para que las operaciones de prueba y depuración resulten más fáciles posteriormente; entre ellos se incluyen:

- Crear un entorno de prueba
- Establecer aserciones
- Mostrar las secuencias de eventos

Crear un entorno de prueba

El entorno de sistema en el que piensa ejecutar una aplicación tiene la misma importancia que el entorno de datos que ha instalado para la propia aplicación. Para garantizar la portabilidad y crear un contexto apropiado para las pruebas y la depuración, debe tener en cuenta lo siguiente:

- El hardware y el software
- Las rutas del sistema y las propiedades de archivo
- La estructura de directorios y las ubicaciones de los archivos

Hardware y software

Para obtener la máxima portabilidad, debe programar las aplicaciones en la plataforma común de menor nivel en la que va a ejecutarlas. Para establecer una plataforma de base:

- Programe las aplicaciones utilizando el modo de vídeo común más bajo.
- Determine los requisitos básicos para la RAM y el espacio de almacenamiento de medios, incluyendo los controladores necesarios o el software que se ejecute de forma simultánea.
- Tenga en cuenta casos especiales de memoria, archivo y bloqueo de registros para las versiones en red y autónoma de las aplicaciones.

Rutas del sistema y propiedades de archivos

Con el fin de garantizar que todos los archivos de programa necesarios son fácilmente accesibles en todos los equipos en los que se ejecuta la aplicación, también puede necesitar una configuración de archivo de base. Para definir una línea de base para la configuración, responda a las siguientes preguntas:

- ¿Necesita su aplicación rutas comunes de sistema?
- ¿Ha establecido propiedades adecuadas de acceso a archivos?
- ¿Se han establecido correctamente permisos de red para cada usuario?

Estructura de directorios y ubicaciones de archivos

Si el código fuente hace referencia a rutas o a nombres de archivos absolutos, es obligatorio que tales rutas y archivos existan en el momento de instalar la aplicación en cualquier otro PC. Para evitar este caso puede seguir uno de estos procedimientos:

- Para obtener información adicional sobre el uso de los archivos de configuración, consulte el capítulo 3, Configurar Visual FoxPro, en la *Guía de instalación.*
- Cree un directorio o una estructura de directorios independiente para mantener los archivos de origen apartados de los archivos de aplicación generados. De esta forma, puede comprobar las referencias de la aplicación terminada y saber exactamente los archivos que necesita distribuir.
- Use rutas de acceso relativas.

Establecer aserciones

Puede incluir aserciones en el código para comprobar las hipótesis que tiene sobre el entorno de ejecución del código.

Para establecer una aserción

- Use el comando ASSERT para identificar las hipótesis del programa.

  Cuando la condición estipulada en el comando ASSERT se evalúa como falsa (.F.), aparece un cuadro de mensajes de aserciones y se repite en la ventana **Resultados del depurador**.

Por ejemplo, podría escribir una función que espera un valor del parámetro distinto de cero. La línea de código siguiente en la función le avisa si el valor del parámetro es 0:

```
ASSERT nParm != 0 MESSAGE "Recibido el parámetro 0"
```

Puede especificar si los mensajes de aserciones se van a mostrar mediante el comando SET ASSERTS. Como valor predeterminado, los mensajes de aserciones no se muestran.

Mostrar las secuencias de eventos

Cuando vea que se producen eventos en relación con otros eventos, puede determinar la ubicación más eficaz para incluir el código.

Para hacer un seguimiento de eventos

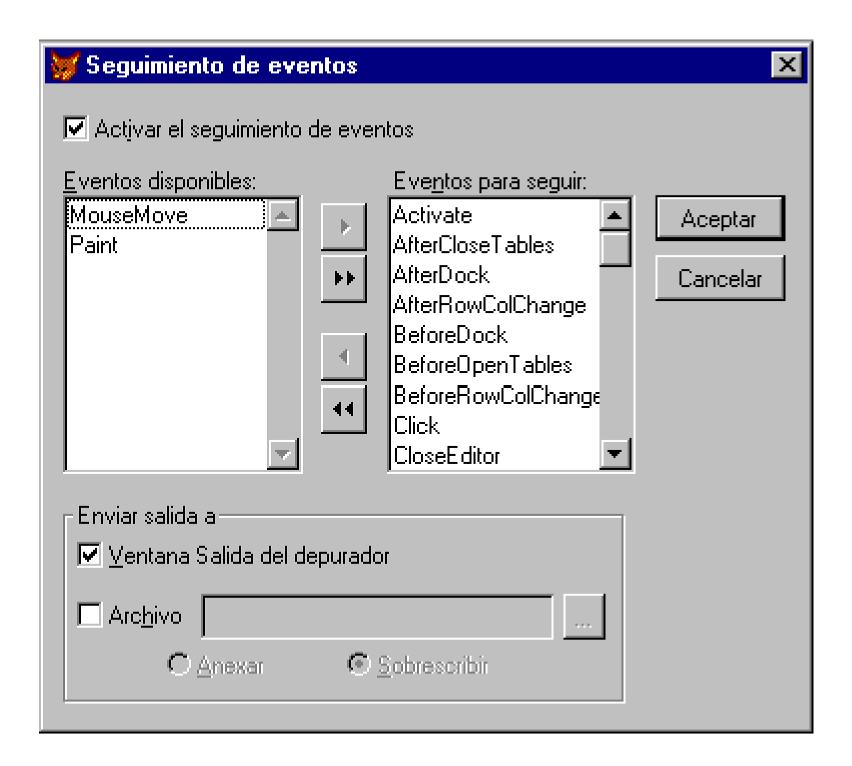
- En el menú **Herramientas** de la ventana Depurador, elija **Seguimiento de eventos**.

  –O bien–

- Use el comando SET EVENTTRACKING.

El cuadro de diálogo **Seguimiento de eventos** permite seleccionar los eventos que desee ver.

Cuadro de diálogo Seguimiento de eventos



Nota En este ejemplo, los eventos MouseMove y Paint se han eliminado de la lista Eventos para seguir, porque estos eventos se producen con tanta frecuencia que dificultan la visión de las secuencias de los demás eventos.

Cuando el seguimiento de eventos está activado, cada vez que se produzca un evento de sistema en la lista Eventos para seguir, el nombre del evento aparecerá en la ventana Resultados del depurador o se escribirá en un archivo. Si opta por mostrar los eventos en la ventana **Resultados del depurador**, también podrá guardarlos en un archivo según se describe en Mostrar los resultados más adelante en este mismo capítulo.

Nota Si la ventana Resultados del depurador no está abierta, los eventos no se presentarán en la lista aun cuando esté activado el cuadro Ventana Resultados del depurador.

Aislar los problemas

Una vez identificados los problemas mediante la prueba, puede usar el entorno de depuración de Visual FoxPro para aislarlos con los pasos siguientes:

- Iniciar una sesión de depuración
- Seguimiento del código
- Suspensión de la ejecución del programa
- Mostrar los valores almacenados
- Mostrar los resultados

Iniciar una sesión de depuración

Para iniciar una sesión de depuración, abra el entorno de depuración.

Para abrir el depurador

- En el menú **Herramientas**, elija **Depurador**.

  Nota Si va a realizar la depuración en el entorno de Visual FoxPro, elija la herramienta de depuración que desee abrir en el menú Herramientas.

También puede abrir el depurador con cualquiera de los comandos siguientes:

DEBUG

SET STEP ON

SET ECHO ON

El depurador se abre automáticamente siempre que se encuentra una condición de punto de interrupción.

Seguimiento del código

Una de las estrategias de depuración más útiles que tiene a su disposición es la posibilidad de realizar un seguimiento del código, ver cada línea de código según se ejecute y comprobar los valores de las variables, propiedades y la configuración del entorno.

Código en la ventana Seguimiento

Seguimiento

Objeto: cmdnext Procedimiento: Clic

```
SKIP 1
IF EOF()
    GO BOTTOM
ENDIF

THIS.Parent.RecordPointerMoved
THIS.Parent.EnableDisableButtons
```

Para hacer el seguimiento del código

1. Inicie una sesión de depuración.
2. Si no hay ningún programa abierto en la ventana Seguimiento, elija **Ejecutar** en el menú **Depurar**.
3. Elija **Paso a paso por instrucciones** en el menú **Depurar** o haga clic en el botón **Paso a paso** de la barra de herramientas.

Una flecha en el área gris situada a la izquierda del código indica la siguiente línea que se va a ejecutar.

Sugerencia Puede aplicar las sugerencias siguientes:

- Establezca puntos de interrupción para reducir el intervalo de código por el que necesita pasar.
- Puede ignorar una línea de código que sepa que va a generar un error colocando el cursor sobre la línea de código después de la línea problemática y eligiendo **Configurar siguiente instrucción** en el menú **Depurar**.
- Si tiene mucho código asociado con eventos Timer, puede evitar el seguimiento de este código desactivando **Mostrar eventos Timer** en la ficha **Depurar** del cuadro de diálogo **Opciones**.

Si aísla un problema cuando esté depurando código de un programa u objeto, podrá repararlo inmediatamente.

Para reparar los problemas encontrados durante el seguimiento del código

- En el menú **Depurar**, elija **Reparar**.

Cuando elija **Reparar** en el menú **Depurar**, la ejecución del programa se cancela y el editor de código se abre donde esté situado el cursor en la ventana **Seguimiento**.

Suspensión de la ejecución del programa

Los puntos de interrupción permiten suspender la ejecución del programa. Una vez suspendida, puede comprobar los valores de las variables y propiedades, ver la configuración del entorno y examinar secciones de código línea a línea sin tener que recorrer todo el código.

Sugerencia También puede suspender la ejecución de un programa que esté en ejecución si presiona la tecla ESC en la ventana **Seguimiento**.

Suspender la ejecución en una línea de código

Puede establecer puntos de interrupción en el código para suspender la ejecución del programa de diferentes maneras. Si conoce el punto exacto en el que desea suspender la ejecución del programa, puede establecer un punto de interrupción directamente en esa línea de código.

Para definir un punto de interrupción en una línea de código determinada

En la ventana Seguimiento, busque la línea de código en la que desee establecer el punto de interrupción y realice una de las operaciones siguientes:

1. Sitúe el cursor en la línea de código.
2. Presione F9 o haga clic en el botón **Alternar punto de interrupción** de la barra de

herramientas **Depurador**.

–O bien–

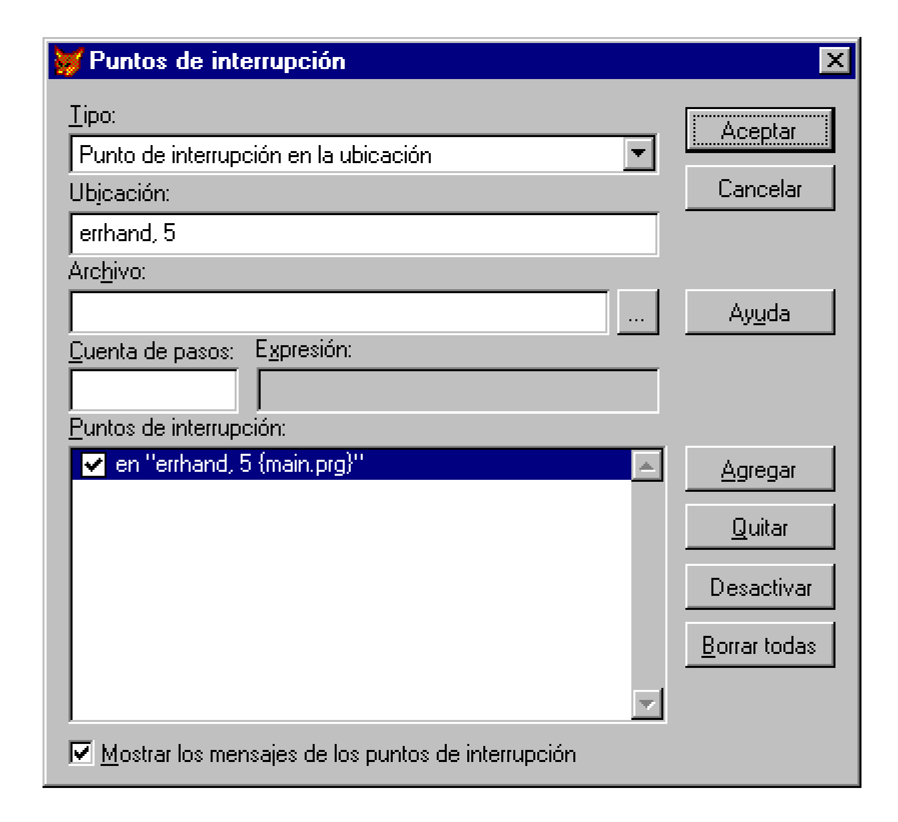
- Haga doble clic en el área gris situada a la izquierda de la línea de código.

Aparecerá un punto sólido en el área gris de la izquierda de la línea de código para indicar que se ha establecido un punto de interrupción en esa línea.

Sugerencia Si va a depurar objetos, puede localizar líneas de código determinadas en la ventana **Seguimiento**; para ello, elija el objeto en la lista Objeto y el método o evento en la lista Procedimiento.

También puede establecer puntos de interrupción si especifica ubicaciones y archivos en el cuadro de diálogo **Puntos de interrupción**.

Punto de interrupción en una ubicación



Ejemplos de ubicaciones y archivos de puntos de interrupción

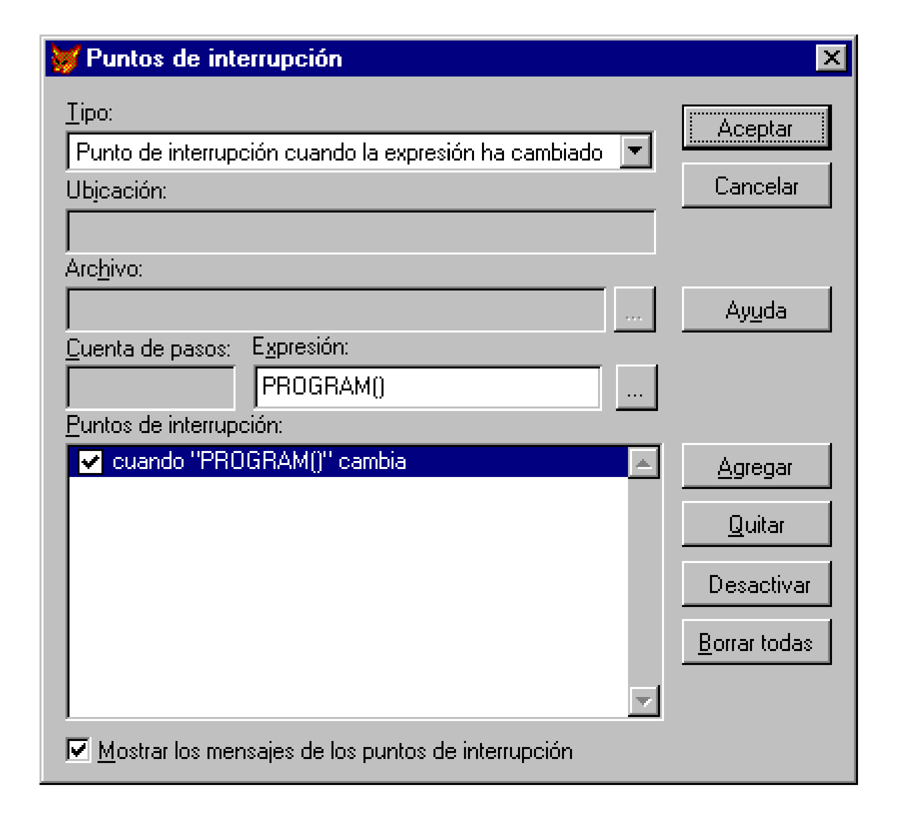
Ubicación	Archivo	Dónde se suspende la ejecución
`ErrHandler`	C:\Myapp\Main.prg	La primera línea ejecutable de un procedimiento denominado `ErrHandler` en Main.prg.
`Main,10`	C:\Myapp\Main.prg	La décima línea del programa

		denominado `Main`.
`Click`	C:\Myapp\Form.scx	La primera línea ejecutable de cualquier procedimiento, función, método o evento denominado `Click` en Form.scx.
`cmdNext.Click`	C:\Myapp\Form.scx	La primera línea ejecutable asociada al evento Click de `cmdNext` en Form.scx.
`cmdNext::Click`		La primera línea ejecutable del evento Click de cualquier control cuya ParentClass sea `cmdNext` en cualquier archivo.

Suspender la ejecución cuando los valores cambian

Si desea saber cuándo cambia el valor de una variable o una propiedad, o cuándo cambia una condición del tiempo de ejecución, puede establecer un punto de interrupción en una expresión.

Punto de interrupción cuando la expresión ha cambiado



Para suspender la ejecución del programa cuando ha cambiado el valor de una expresión

1. En el menú **Herramientas** de la ventana Depurador, elija **Puntos de interrupción** para abrir el cuadro de diálogo Puntos de interrupción.

2. En la lista **Tipo**, elija **Punto de interrupción cuando la expresión ha cambiado**.

3. Escriba la expresión en el cuadro **Expresión**.

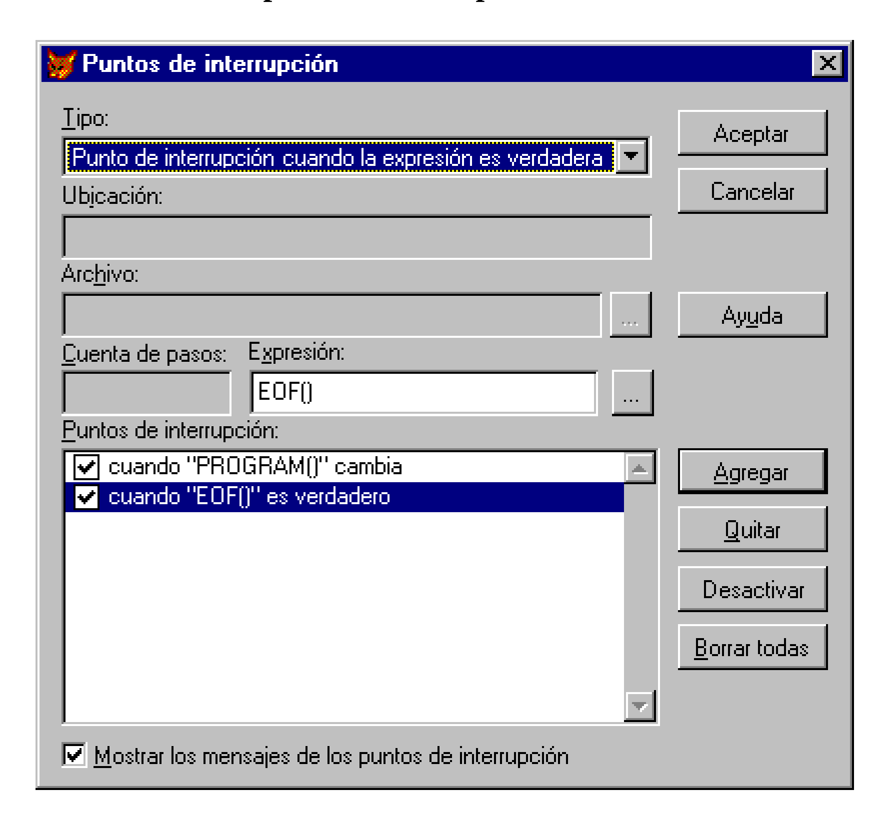
Ejemplos de expresiones de puntos de interrupción

Expresión	Uso
`RECNO( )`	Suspender la ejecución cuando el puntero de registro se mueve en la tabla.
`PROGRAM( )`	Suspender la ejecución en la primera línea de cualquier programa, procedimiento, método o evento nuevo.
`myform.Text1.Value`	Suspender la ejecución siempre que se cambie el valor de esta propiedad de forma interactiva o mediante programación.

Suspender de forma condicional la ejecución

A menudo deseará suspender la ejecución de un programa no en una línea determinada, sino cuando una determinada condición sea verdadera.

Punto de interrupción en una expresión



Para suspender la ejecución del programa cuando una expresión se evalúe como verdadera

1. En el menú **Herramientas** de la ventana Depurador, elija **Puntos de interrupción** para abrir el

cuadro de diálogo Puntos de interrupción.

2. En la lista **Tipo**, elija **Punto de interrupción cuando la expresión es verdadera**.

3. Escriba la expresión en el cuadro **Expresión**.

4. Elija **Agregar** para agregar el punto de interrupción a la lista **Puntos de interrupción**.

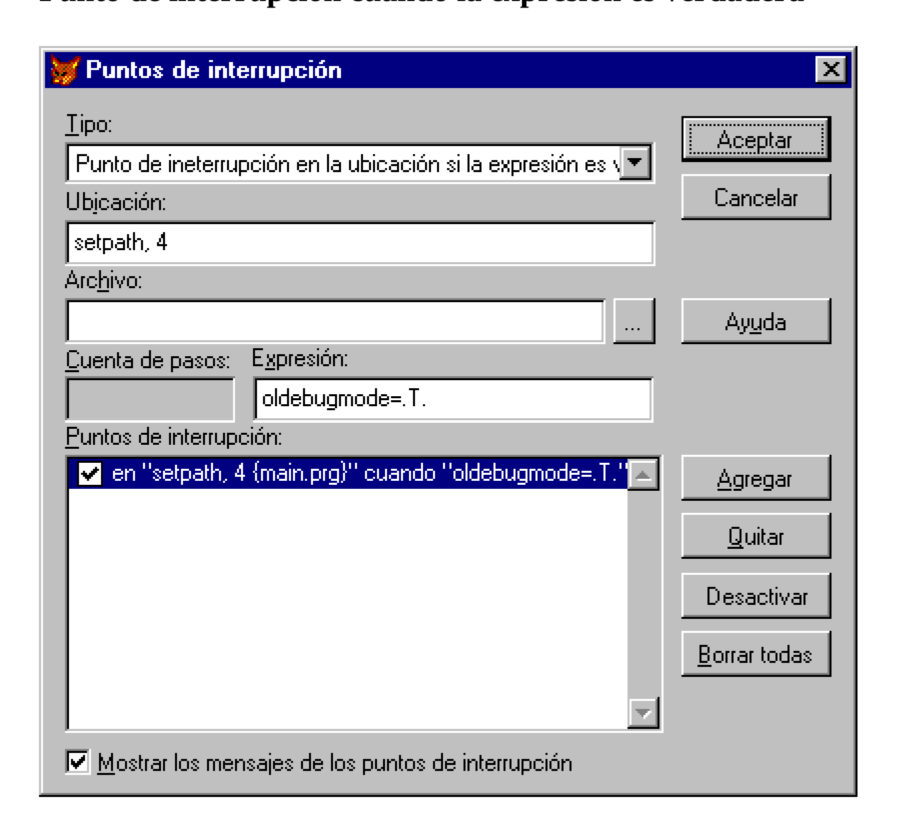
Ejemplos de expresiones de puntos de interrupción

Expresión	Uso
`EOF( )`	Suspender la ejecución cuando el puntero de registro se ha desplazado detrás del último registro de una tabla.
`'CLICK'$PROGRAM( )`	Suspender la ejecución de la primera línea de código asociado a un evento Click o DblClick.
`nReturnValue = 6`	Si el valor devuelto de un cuadro de mensajes se almacena en `nReturnValue`, se suspende la ejecución cuando un usuario elige **Sí** en el cuadro de mensajes.

Suspender de forma condicional la ejecución en una línea de código

Puede especificar que la ejecución del programa se suspenda en una línea determinada sólo cuando una condición concreta sea verdadera.

Punto de interrupción cuando la expresión es verdadera



Para suspender la ejecución del programa en una línea determinada cuando la expresión se evalúa como verdadera

1. En el menú **Herramientas** de la ventana Depurador, elija **Puntos de interrupción** para abrir el cuadro de diálogo Puntos de interrupción.

2. En la lista **Tipo**, elija **Punto de interrupción cuando la expresión es verdadera**.

3. Escriba la ubicación en el cuadro **Ubicación**.

4. Escriba la expresión en el cuadro **Expresión**.

5. Elija **Agregar** para agregar el punto de interrupción a la lista **Puntos de interrupción**.

6. Elija **Aceptar**.

   Sugerencia Suele ser más fácil buscar la línea de código en la ventana Seguimiento, establecer un punto de interrupción y modificarlo en el cuadro de diálogo Punto de interrupción. Para ello, cambie el **Tipo** de **Punto de interrupción en la ubicación** a **Punto de interrupción cuando la expresión es verdadera** y, a continuación, agregue la expresión.

Quitar puntos de interrupción

Puede desactivar los puntos de interrupción sin quitarlos en el cuadro de diálogo **Puntos de interrupción**. Puede eliminar puntos de interrupción del tipo "punto de interrupción en la ubicación" en la ventana **Seguimiento**.

Para quitar un punto de interrupción de una línea de código

En la ventana Seguimiento, busque el punto de interrupción y realice una de las operaciones siguientes:

- Sitúe el cursor en la línea de código y elija **Alternar punto de interrupción** en la barra de herramientas **Depurador.**

  –O bien–

- Haga doble clic en el área gris situada a la izquierda de la línea de código.

Mostrar los valores almacenados

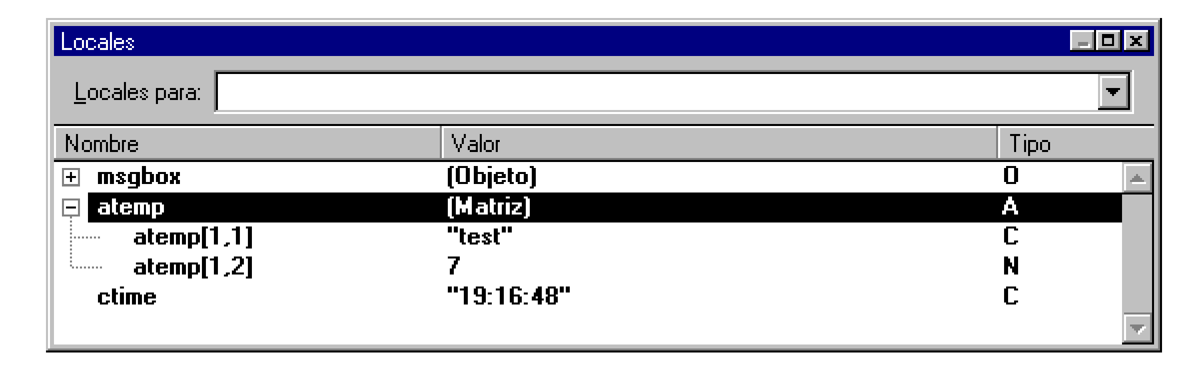
En la ventana Depurador, puede ver fácilmente los valores de tiempo de ejecución de las variables, elementos de matriz, propiedades y expresiones en las ventanas siguientes:

- Ventana Locales
- Ventana Inspección
- Ventana Seguimiento

Mostrar los valores almacenados en la ventana Locales

La ventana Locales muestra todas las variables, matrices, objetos y miembros de objetos que están visibles en cualquier programa, procedimiento o método de la pila de llamadas. Como opción predeterminada, los valores del programa que se está ejecutando actualmente se muestran en la ventana Locales. Puede ver estos valores para otros programas o procedimientos de la pila de llamadas si elige los programas o procedimientos en la lista Locales para.

Ventana Locales



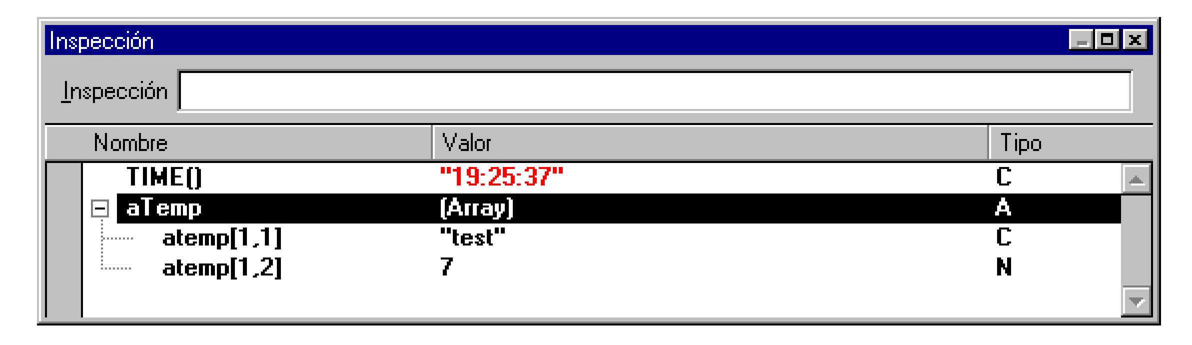
Puede efectuar un análisis descendente de matrices u objetos si hace clic en el signo más (+) situado junto al nombre de la matriz u objeto en las ventanas Locales e Inspección. Cuando efectúe el análisis descendente, podrá ver los valores de todos los elementos de las matrices y las configuraciones de propiedades de los objetos.

Puede incluso cambiar los valores de las variables, elementos de matrices y propiedades en las ventanas Locales e Inspección seleccionando la variable, elemento de matriz o propiedad si hace clic en la columna Valor y escribe un valor nuevo.

Mostrar los valores almacenados en la ventana Inspección

En el cuadro Inspección de la ventana Inspección, escriba cualquier expresión válida de Visual FoxPro y presione ENTRAR. El valor y el tipo de expresión aparecen en la lista de la ventana Inspección.

Ventana Inspección



Nota En la ventana Inspección no se pueden introducir expresiones que creen objetos.

También puede seleccionar variables o expresiones en la ventana Seguimiento o en otras ventanas del Depurador y arrastrarlas hasta la ventana Inspección.

Los valores que se hayan modificado aparecen en rojo en la ventana Inspección.

Para quitar un elemento de la lista de la ventana Inspección

Seleccione el elemento y elija una de las opciones siguientes:

- Presione SUPR.

  –O bien–

- En el menú contextual, elija **Eliminar inspección**.

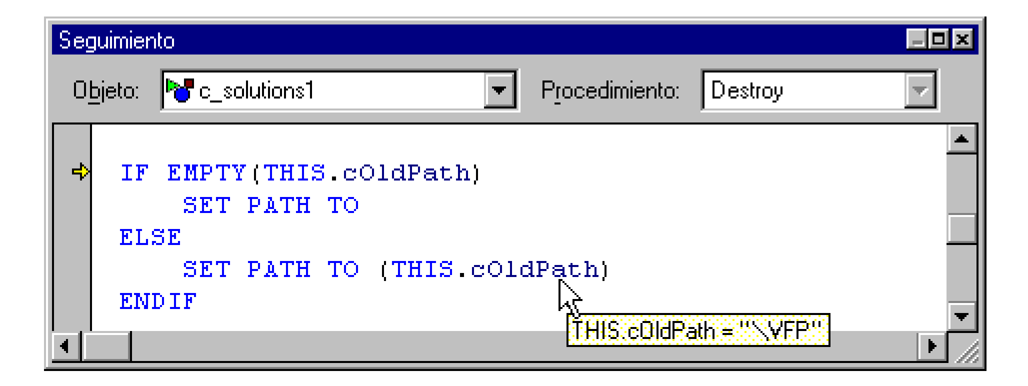
Para modificar una inspección

- Haga doble clic en la inspección en la ventana **Inspección** y modifíquela.

Mostrar los resultados en la ventana Seguimiento

Sitúe el cursor sobre cualquier variable, elemento de matriz o propiedad de la ventana Seguimiento para mostrar su valor actual en una sugerencia del valor.

Sugerencia del valor en la ventana Seguimiento



Mostrar los resultados

El comando DEBUGOUT permite escribir valores de la ventana Resultados del depurador en un registro de archivo de texto. Además puede usar el comando SET DEBUGOUT TO o la ficha Depurar del cuadro de diálogo Opciones.

Si no está escribiendo comandos DEBUGOUT en un archivo de texto, la ventana Resultados del depurador debe estar abierta para que pueda escribir los valores de DEBUGOUT. La siguiente línea de código se imprime en la ventana Resultados del depurador en el momento en que se ejecuta la línea de código:

```
DEBUGOUT DATETIME( )
```

Además, puede activar el seguimiento de eventos, que se describe en apartados anteriores de este capítulo y optar por hacer que el nombre y los parámetros de cada evento que se produzca se muestren en la ventana Resultados del depurador.

Registrar el trayecto del código

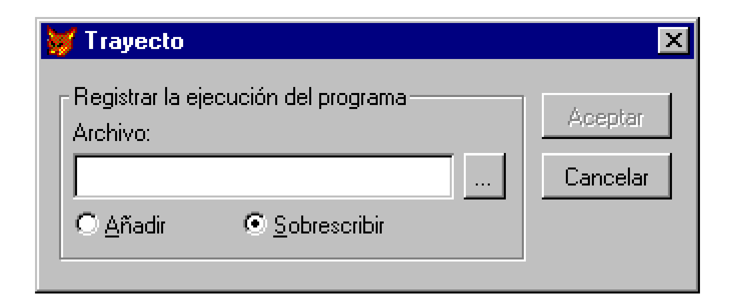
En un proceso de programación posterior, podría perfeccionar el código para obtener más rendimiento y garantizar que lo ha probado correctamente mediante un registro de la información de trayecto del código.

El trayecto del código ofrece información sobre las líneas de código que se han ejecutado y el tiempo empleado para ejecutarlas. Esta información puede ayudarle a identificar las áreas de código que no se están ejecutando y, por tanto, tampoco se están comprobando, así como las que podría modificar para obtener mayor rendimiento.



Puede activar y desactivar el trayecto del código si hace clic en el botón Trayecto de código de la barra de herramientas de la ventana Depurador. Si activa el trayecto de código, se abre el cuadro de diálogo Trayecto donde puede especificar un archivo en el que se guardará esta información.

Cuadro de diálogo Trayecto



También puede activar o desactivar el registro de trayecto mediante programación mediante el comando SET COVERAGE TO. Por ejemplo, podría incluir el comando siguiente en la aplicación inmediatamente antes de la parte de código que desea investigar:

```
SET COVERAGE TO miregistro.log
```

Después de la sección de código cuyo trayecto desee registrar, podría incluir el comando siguiente para desactivar el trayecto de código:

```
SET COVERAGE TO
```

Una vez especificado un archivo para la información de trayecto, cambie a la ventana principal de Visual FoxPro y ejecute el programa, formulario o aplicación. En el archivo de registro se escribe la información siguiente para cada línea de código que se ejecute:

- Tiempo, en segundos, empleado en la ejecución de la línea.
- La clase a la que pertenece el código.
- El método o procedimiento en el que se encuentra la línea.
- El número de la línea de código.
- El archivo en el que se encuentra el código.

La forma más sencilla de extraer información del archivo de registro es convertirla en una tabla de modo que se puedan establecer filtros, ejecutar consultas e informes, ejecutar comandos y manipular la tabla de otras maneras.

La aplicación Analizador de trayecto crea un cursor a partir de los datos generados en el registro de trayecto y lo utiliza en una ventana para facilitar el análisis.

El programa siguiente convierte en una tabla el archivo de texto creado por el registro de trayecto:

```
cFileName = GETFILE('DBF')
IF EMPTY(cFileName)
   RETURN
ENDIF

CREATE TABLE (cFileName) ;
   (duration n(7,3), ;
   class c(30), ;
   procedure c(60), ;
   line i, ;
   file c(100))

APPEND FROM GETFILE('log') TYPE DELIMITED
```

Controlar errores de tiempo de ejecución

Los errores de tiempo de ejecución se producen después de comenzar la ejecución de la aplicación. Entre las acciones que podrían generar errores de tiempo de ejecución se encuentran la escritura en un archivo que no existe, el intento de abrir una tabla que ya está abierta, el intento de seleccionar una tabla que se ha cerrado, la presencia de un conflicto de datos, la división de un valor entre cero, etc.

Los comandos y funciones siguientes resultan útiles para anticiparse y gestionar los errores de tiempo de ejecución.

Para	**Use**
Llenar una matriz con información de error	AERROR( )
Abrir la ventana Depurador o Seguimiento	DEBUG o SET STEP ON
Generar un error específico para probar el control de errores	ERROR
Devolver un número de error	ERROR( )
Devolver una línea de programa en ejecución	LINENO( )

Devolver una cadena de mensaje de error	MESSAGE( )
Ejecutar un comando cuando se produce un error	ON ERROR
Devolver comandos asignados a los comandos de control de errores	ON( )
Devolver el nombre del programa actualmente en ejecución	PROGRAM( ) o SYS(16)
Volver a ejecutar el comando anterior	RETRY
Devolver cualquier parámetro de mensaje de error actual	SYS(2018)

Previsión de errores

La primera línea de defensa contra errores de tiempo de ejecución es anticiparse a que ocurran y codificarlos. Por ejemplo, esta línea de código mueve el puntero al siguiente registro de la tabla:

```
SKIP
```

Este código funciona a menos que el puntero de registro ya esté situado detrás del último registro de la tabla y es en este punto donde se producirá el error.

Las líneas de código siguiente se anticipan a este error y lo evitan:

```
IF !EOF()
   SKIP
      IF EOF()
         GO BOTTOM
      ENDIF
ENDIF
```

Otro ejemplo es esta línea de código que muestra el cuadro de diálogo **Abrir** para permitir al usuario abrir una tabla en un área de trabajo nueva:

```
USE GETFILE('DBF') IN 0
```

El problema es que el usuario podría elegir **Cancelar** en el cuadro de diálogo **Abrir** o escribir el nombre de un archivo que no existe. El código siguiente se anticipa a esta situación comprobando que el archivo existe antes de que el usuario intente utilizarlo:

```
IF FILE(cNewTable)
   USE (cNewTable) IN 0
ENDIF
```

El usuario final también podría escribir el nombre de un archivo que no sea una tabla de Visual FoxPro. Para solucionar este problema, podría abrir el archivo con las funciones de E/S de archivo a bajo nivel, analizar el encabezado binario y comprobar que el archivo es una tabla válida. Sin embargo, esto implica un pequeño trabajo y podría ralentizar perceptiblemente la aplicación. Lo mejor sería controlar la situación durante el tiempo de ejecución mostrando un mensaje como "Abra

otro archivo. Este archivo no es una tabla". cuando se produzca el error 15, "No es una tabla".

No puede, y probablemente tampoco lo desee, anticipar todos los errores posibles, por lo que tendrá que interrumpir algunos mediante código que se ejecutará en caso de que se produzca un error de tiempo de ejecución.

Controlar errores de procedimientos

Cuando se produzca un error en el código de procedimientos, Visual FoxPro comprobará el código de control de errores asociado a una rutina ON ERROR. Si no existe ninguna rutina ON ERROR, Visual FoxPro muestra el mensaje de error predeterminado de Visual FoxPro. Para obtener una lista completa de los mensajes de error de Visual FoxPro y los números de error, vea la Ayuda.

Crear una rutina ON ERROR

Puede incluir cualquier comando o expresión válida de FoxPro después de ON ERROR, pero normalmente se llama a un procedimiento o programa de control de errores.

Para ver el funcionamiento de ON ERROR, puede escribir en la ventana Comandos un comando irreconocible, como:

```
qxy
```

Aparecerá un cuadro de diálogo de mensaje de error estándar de Visual FoxPro que indica "No se reconoce el verbo de comando". No obstante, si ejecuta las líneas de código siguientes, aparecerá el error número 16 impreso en la ventana de salida activa en lugar de mostrarse el mensaje de error estándar en un cuadro de diálogo:

```
ON ERROR ?ERROR()
qxy
```

Si se ejecuta ON ERROR sin escribir nada más detrás, se restablecen los mensajes de error incorporados de Visual FoxPro:

```
ON ERROR
```

De forma esquemática, el código siguiente ilustra un controlador de errores ON ERROR:

```
LOCAL lcOldOnError

* Guardar el controlador de errores original
lcOldOnError = ON("ERROR")

* Ejecutar ON ERROR con el nombre de un procedimiento
ON ERROR DO errhandler WITH ERROR(), MESSAGE()

* código al que se aplica la rutina de control de errores

* Restablecer el controlador de errores original
ON ERROR &lcOldOnError

PROCEDURE errhandler
LOCAL aErrInfo[1]
```

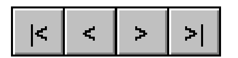
```
AERROR(aErrInfo)
DO CASE
   CASE aErrInfo[1] = 1 && El archivo no existe
      * mostrar el mensaje apropiado
      * y emprender alguna acción para solucionar el problema.
   OTHERWISE
      * mostrar un mensaje genérico, y quizá enviar
      * un mensaje electrónico de alta prioridad al administrador
ENDPROC
```

Controlar errores en clases y objetos

Cuando se produce un error en el código del método, Visual FoxPro comprueba el código de gestión de errores asociado al evento Error del objeto. Si no se ha escrito código a nivel del objeto para el evento Error, se ejecuta el código del evento Error heredado de la clase principal o de otra clase superior de la jerarquía de clases. Si no se ha escrito código para el evento Error en ninguna clase de la jerarquía, Visual FoxPro comprueba la existencia de una rutina ON ERROR. Si no existiera, Visual FoxPro mostrará el mensaje de error predeterminado de Visual FoxPro.

En las clases puede encapsularse todo lo que necesita un control, incluido el control de errores para que el control pueda utilizarse en diversos entornos. Si posteriormente descubre otro error que podría encontrar el control, puede agregar la gestión de ese error a la clase; de este modo, todos los objetos basados en la clase heredarán automáticamente la función de control del nuevo error.

Por ejemplo, la clase `vcr` de la biblioteca de clases Buttons.vcx, ubicada en el directorio ...\Samples\Vfp98\Classes de Visual Studio se basa en la clase de contenedor de Visual FoxPro.



Cuatro botones de comandos del contenedor controlan el desplazamiento por la tabla moviendo el puntero de registro en una tabla con los comandos siguientes:

```
GO TOPSKIP - 1SKIP 1GO BOTTOM.
```

Podría producirse un error si un usuario elige uno de los botones y no hay ninguna tabla abierta. Visual FoxPro intenta escribir en una tabla valores almacenados en el búfer cuando se mueve el puntero de registro. Por tanto, podría producirse un error si se activa el almacenamiento optimista de filas en búfer y otro usuario ha cambiado un valor en el registro almacenado en el búfer.

Estos errores podrían producirse cuando el usuario elige cualquiera de los botones; por tanto, no tiene sentido tener cuatro métodos diferentes de control de errores. El código siguiente asociado al evento Error de cada uno de los botones de comando transfiere la información del error a una sola rutina de gestión de errores de la clase:

```
LPARAMETERS nError, cMethod, nLine
THIS.Parent.Error(nError, cMethod, nLine)
```

El código siguiente está asociado al evento Error de la clase `vcr`. El código real es diferente debido a los requisitos de codificación para la localización.

```
Parameters nError, cMethod, nLine
DO CASE
```

```
CASE nError = 13 && No se ha encontrado el alias
   cNewTable = GETFILE('DBF')
   IF FILE(cNewTable)
      SELECT 0
      USE (cNewTable)
      This.SkipTable = ALIAS()
   ELSE
      This.SkipTable = ""
   ENDIFCASE nError = 1585 && Conflicto de datos
* Actualizar conflicto controlado por una clase de comprobación de datos
   nConflictStatus = ;
      THIS.DataChecker1.CheckConflicts()
   IF nConflictStatus = 2
      MESSAGEBOX "No se puede resolver un conflicto de datos."
   ENDIFOTHERWISE
* Mostrar información sobre otros errores.

   cMsg="Error:" + ALLTRIM(STR(nError)) + CHR(13) ;
      + MESSAGE()+CHR(13)+"Programa:"+PROGRAM()

   nAnswer = MESSAGEBOX(cMsg, 2+48+512, "Error")
   DO CASE
      CASE nAnswer = 3   &&Anular
         CANCEL
      CASE nAnswer = 4   &&Reintentar
         RETRY
      OTHERWISE       && Ignorar
         RETURN
   ENDCASEENDCASE
```

Es posible que desee estar seguro de que suministra información para un error que nunca ha controlado. De lo contrario, el código del evento de error se ejecutará pero no realizará ninguna acción y ya no se mostrará el mensaje de error predeterminado de Visual FoxPro. El resultado es que ni el programador ni el usuario sabrán lo que ha sucedido.

Dependiendo de los usuarios a los que esté destinada la aplicación, podría suministrar más información en el caso de un error no controlado, como el nombre y número de teléfono de la persona a la que se puede solicitar ayuda.

Retorno desde el código de control de errores

Después de ejecutarse el código de control de errores, se ejecuta la línea de código que sigue a la que ha producido el error. Si desea volver a ejecutar la línea de código que ha producido el error cuando haya cambiado la situación que lo ha producido, use el comando RETRY.

Nota Puede llamar al evento Error cuando el error encontrado no esté asociado a una línea del código. Por ejemplo, si llama al método CloseTables de un entorno de datos cuando AutoCloseTables está establecido como verdadero (.T.) y se libera el formulario, se generará un error interno cuando Visual FoxPro intente volver a cerrar las tablas. Puede interrumpir en este error pero no existe ninguna línea de código para ejecutar RETRY.

Capítulo 15: Optimizar aplicaciones

Cuando utilice Visual FoxPro para diseñar y ejecutar aplicaciones, querrá obtener el máximo rendimiento tanto de su sistema operativo como de Visual FoxPro y de su aplicación.

Para obtener información sobre la forma de obtener el máximo rendimiento del equipo y del sistema operativo, consulte el capítulo 4, Optimizar el sistema, de la *Guía de instalación e Índice principal*.

En este capítulo se describe:

- Optimizar tablas e índices
- Usar Rushmore para agilizar el acceso a datos
- Optimizar formularios y controles
- Optimizar programas
- Optimizar controles ActiveX
- Optimizar aplicaciones en entornos multiusuario
- Optimizar el acceso a datos remotos
- Optimizar aplicaciones internacionales

Optimizar tablas e índices

Puede agilizar el acceso a datos de tablas mediante índices y almacenamiento en búfer de forma eficaz. Además, puede usar la tecnología Rushmore para optimizar las consultas.

Usar índices

El uso de índices permite agilizar el acceso a los datos de una tabla. Si se agrega un índice a una tabla se agilizan las búsquedas, especialmente si se puede utilizar la tecnología Rushmore para optimizar la búsqueda. El uso de índices también permite trabajar con los datos en un orden determinado, como ver una tabla de clientes ordenada por apellido.

Si los registros de una tabla tienen claves exclusivas, cree un índice principal o candidato en el campo. Estos tipos de índice permiten que Visual FoxPro valide la tecla a un nivel inferior con lo que se consigue el máximo rendimiento.

Además de usar índices en los campos utilizados para buscar y ordenar, también debería indexar todos los campos relacionados con una combinación. Si combina dos tablas en campos que no están indexados, la operación de combinación puede tardar mucho más tiempo en realizarse.

Una característica importante de Visual FoxPro es que puede crear un índice en cualquier expresión (en algunas bases de datos sólo puede indexar los campos). Esta capacidad permite usar índices para optimizar las operaciones de búsqueda, ordenación o combinación en conjuntos de campos o en expresiones derivadas de campos. Por ejemplo, puede crear un índice de un campo de nombres basado en una expresión que use la función SOUNDEX( ). De esa manera, la aplicación puede proporcionar un acceso muy rápido a los nombres que guarden un cierto parecido.

Al agregar índices a las tablas, debe ver si la mejora en los tiempos de recuperación compensa la pérdida de rendimiento al actualizar la tabla. A medida que agregue más índices a la tabla, las actualizaciones e inserciones en la tabla se ralentizarán porque Visual FoxPro necesita actualizar cada índice.

Por último, evite el uso de índices en campos que contengan sólo unos cuantos valores discretos,

como los campos lógicos. En estos casos, el índice sólo contiene un pequeño número de entradas y el trabajo de mantener el índice probablemente sea mayor que la ventaja que se consigue a la hora de realizar búsquedas.

Para obtener detalles sobre la forma de crear índices de forma eficaz al usar la tecnología Rushmore, vea Usar la tecnología Rushmore para agilizar el acceso a los datos, más adelante en este capítulo.

Optimizar combinaciones

Cuando vaya a crear combinaciones mediante SELECT - SQL, las situaciones siguientes pueden reducir el rendimiento y producir resultados imprevistos:

- Combinación de tablas en datos que no sean una clave principal o exclusiva en una de las tablas.
- Combinación de tablas que contengan campos vacíos.

Para evitar estas situaciones, cree combinaciones que se basen en la relación entre las claves principales de una tabla y las claves externas de la otra. Si crea una combinación basada en datos que no sean exclusivos, el resultado final puede ser el producto de las dos tablas. Por ejemplo, la instrucción SELECT - SQL siguiente crea una combinación que puede producir un resultado muy grande:

```
SELECT *;
 FROM  tastrade!customer INNER JOIN tastrade!orders ;
   ON  Customer.postal_code = Orders.postal_code
```

En el ejemplo, el código postal identifica de forma exclusiva una ubicación dentro de una ciudad pero tiene poco valor si lo que pretende es establecer una correspondencia entre las filas de clientes y las filas de sus pedidos. El código postal no identifica necesariamente un cliente o un pedido de forma exclusiva. En su lugar, cree una combinación con una instrucción como la siguiente:

```
SELECT *;
 FROM  tastrade!customer INNER JOIN tastrade!orders ;
   ON  Customer.customer_id = Orders.customer_id
```

En este ejemplo, el campo `customer_id` identifica de forma exclusiva un cliente determinado y los pedidos que pertenecen a ese cliente y, por tanto, crea un conjunto de resultados que combina la fila de clientes con cada fila de pedidos.

Además, tenga cuidado al combinar tablas que tengan campos vacíos porque Visual FoxPro establecerá una correspondencia con los campos vacíos. Sin embargo, no establece una correspondencia con los campos que contengan el valor nulo. Al crear una combinación, califique las expresiones del campo en la condición de combinación probando con una cadena vacía.

Por ejemplo, si piensa que el campo de identificación del cliente de la tabla Orders podría estar vacío, use una instrucción como la siguiente para filtrar los registros de pedidos que no tengan número de cliente:

```
SELECT *;
 FROM  tastrade!customer INNER JOIN tastrade!orders ;
   ON  Customer.customer_id = Orders.customer_id; (error w/o ;)
```

```
WHERE tastrade!orders <> ""
```

Sugerencia También puede probar una cadena vacía con la función EMPTY( ), pero la inclusión de una llamada a la función dentro de la expresión de filtro no resulta tan rápida como efectuar una comparación con un valor constante.

Usar el Administrador de proyectos

Cuando use el Administrador de proyectos, puede combinar un número ilimitado de programas y procedimientos en un solo archivo .app o .exe. Con esto se aumenta considerablemente la velocidad de ejecución del programa por varias razones.

En primer lugar, Visual FoxPro abre un archivo de programa y lo deja abierto. Después, cuando se ejecuta un comando DO en un programa contenido en el archivo, Visual FoxPro no necesita abrir un archivo adicional.

En segundo lugar, una aplicación de uno o dos archivos reduce el número de archivos necesarios en el directorio de trabajo. La velocidad de todas las operaciones con el archivo aumentará a medida que el sistema operativo tenga menos entradas de directorios que examinar al abrir, cambiar el nombre o eliminar archivos.

Para obtener más información sobre el uso del Administrador de proyectos para crear aplicaciones, consulte el capítulo 13, Compilar una aplicación.

Sugerencias generales para la optimización de tablas e índices

Para crear tablas e índices de la forma más rápida posible, siga las recomendaciones que se muestran a continuación.

- Si no está activado el almacenamiento de registros o tablas en búfer, use INSERT - SQL en lugar de APPEND BLANK seguido por REPLACE, especialmente con una tabla indexada en un entorno multiusuario, porque los índices deben actualizarse una sola vez.
- Si necesita anexar un gran número de registros a una tabla indexada, sería más rápido quitar el índice, anexar los registros y después volver a crear el índice.
- En las instrucciones SQL, evite las llamadas a funciones en la medida de lo posible, especialmente en instrucciones que devuelvan más de un registro, porque la instrucción debe volverse a evaluar (lo que implica una nueva llamada a las funciones) para cada registro. Si crea una instrucción SQL con datos variables, use las expresiones de nombre o la sustitución de macros en lugar de la función EVALUATE( ). Una estrategia aún mejor es crear toda la instrucción de forma dinámica, no en cláusulas individuales. Para obtener más información, vea Usar macros y Crear expresiones de nombre, en la Ayuda.
- Si suele usar un orden de índice determinado, puede mejorar el rendimiento si ordena periódicamente la tabla en este orden.
- Use archivos .CDX en lugar de .IDX en entornos multiusuario porque podrá actualizar un solo archivo .CDX con más rapidez que si actualiza múltiples archivos .IDX.

Usar Rushmore para agilizar el acceso a los datos

Para ayudarle a optimizar el rendimiento de las aplicaciones, Visual FoxPro incluye la tecnología

Rushmore para acceso a los datos. Con Rushmore puede ejecutar determinadas operaciones complejas con tablas muchísimo más rápido que sin esta tecnología.

Explicar la tecnología Rushmore

La tecnología Rushmore es una técnica de acceso a datos que usa los índices estándar de Visual FoxPro para optimizar el acceso a datos. Puede utilizar Rushmore con cualquier índice de Visual FoxPro, incluyendo los índices de FoxPro 1.x (.idx), los índices compactos (.idx) o los índices compuestos (.cdx).

Tanto los índices .cdx como los índices compactos .idx utilizan una técnica de compresión que genera índices de hasta 1/16 del tamaño de los índices de formato tradicional. Visual FoxPro puede procesar índices compactos con mayor rapidez porque son físicamente más pequeños. Esto significa que Visual FoxPro requiere menos acceso a disco para procesar un índice y puede almacenar localmente una parte mayor del índice. Aunque Rushmore, como otras técnicas de acceso a archivos, se beneficia del menor tamaño de los índices compactos, también funciona correctamente con los índices de formatos antiguos.

Cuando Visual FoxPro procesa tablas muy grandes en equipos con poca memoria, es posible que Rushmore no disponga de suficiente memoria para operar. En este caso, aparecerá un mensaje de advertencia ("No hay suficiente memoria para la optimización"). Su programa funcionará correctamente sin perder ningún dato; sin embargo, la consulta no se beneficiará de la tecnología Rushmore.

En su forma más sencilla, Rushmore incrementa la velocidad de los comandos en operaciones con una sola tabla mediante cláusulas FOR que especifican los conjuntos de registros en función de los índices existentes. Además, Rushmore puede aumentar la velocidad de funcionamiento de algunos comandos, como LOCATE e INDEX. Para ver una lista completa de los comandos optimizables, consulte la sección "Usar Rushmore con tablas".

Los comandos SQL de Visual FoxPro utilizan la tecnología Rushmore como herramienta básica para la optimización de consultas de múltiples tablas, utilizando los índices existentes e incluso creando nuevos índices con fines específicos para aumentar la velocidad de dichas consultas.

Usar Rushmore con tablas

Utilice Rushmore para optimizar el acceso a datos según el número de tablas que intervengan. Cuando tenga acceso a una única tabla, podrá beneficiarse de Rushmore siempre que aparezca una cláusula FOR. Cuando tenga acceso a múltiples tablas, las consultas SELECT - SQL prevalecen sobre todas las optimizaciones Rushmore. En un comando SQL, Visual FoxPro decide lo que se necesita para optimizar una consulta y realiza las operaciones automáticamente. No es necesario que abra tablas o índices. Si SQL decide que necesita índices, crea índices temporales para su propio uso.

Para utilizar Rushmore

Elija una de las siguientes opciones:

- Para tener acceso a datos que estén contenidos en una única tabla, utilice una cláusula FOR en un comando como AVERAGE, BROWSE o LOCATE o use comandos SQL para actualizar

tablas. Para obtener una lista completa de comandos que utilizan la cláusula FOR, consulte la tabla siguiente.

–O bien–

- Para tener acceso a datos contenidos en más de una tabla, utilice un comando SELECT - SQL, DELETE - SQL y UPDATE - SQL.

La tabla siguiente muestra una relación de los comandos que utilizan cláusulas FOR. Rushmore está diseñado para que su velocidad sea proporcional al número de registros obtenidos en la operación.

Comandos que se pueden optimizar con cláusulas FOR

AVERAGE	BLANK
BROWSE	CALCULATE
CHANGE	COPY TO
COPY TO ARRAY	COUNT
DELETE	DISPLAY
EDIT	EXPORT TO
INDEX	JOIN WITH
LABEL	LIST
LOCATE	RECALL
REPLACE	REPLACE FROM ARRAY
REPORT	SCAN
SET DELETED	SET FILTER
SORT TO	SUM
TOTAL TO	

Si utiliza una cláusula de alcance, además de una expresión de cláusula FOR optimizable, el alcance debe ser ALL o REST para poder aprovechar la tecnología Rushmore. Las cláusulas de alcance NEXT o RECORD desactivan Rushmore. Puesto que el alcance predeterminado es ALL, Rushmore funcionará cuando omita la cláusula de alcance.

Rushmore puede utilizar cualquiera de los índices abiertos salvo los índices filtrados y UNIQUE.

Nota Para obtener un rendimiento óptimo, no establezca el orden de la tabla.

Si crea índices o etiquetas, el orden se establecerá automáticamente. Si desea sacar el máximo rendimiento de Rushmore a la hora de manejar un conjunto de datos masivo con un orden

determinado, ejecute SET ORDER TO para desactivar el control de los índices y luego utilice el comando SORT.

Indexado eficaz para Rushmore

Rushmore no puede aplicarse a todos los índices. Si usa una cláusula FOR en el comando INDEX, Rushmore no puede utilizar el índice para la optimización. Por ejemplo, debido a que contiene una cláusula FOR, la instrucción siguiente no puede optimizarse:

```
INDEX ON ORDNUM FOR DISCOUNT > 10 TAG ORDDISC
```

De forma análoga, Rushmore no puede usar un índice creado con una *condición* NOT. Por ejemplo, la instrucción siguiente puede optimizarse:

```
INDEX ON DELETED() TAG DEL
```

Sin embargo, ésta no puede optimizarse:

```
INDEX ON NOT DELETED() TAG NOTDEL
```

En el caso especial de que desee excluir los registros eliminados de una consulta, use un índice como el del primer ejemplo para agilizar las operaciones cuando haya establecido SET DELETED como ON.

Funcionamiento sin Rushmore

Las operaciones de obtención de datos se ejecutan sin la optimización Rushmore en las siguientes situaciones:

- Cuando Rushmore no puede optimizar la expresión de la cláusula FOR en un comando posiblemente optimizable. Para más información sobre la creación de expresiones optimizables FOR, consulte la sección Combinar expresiones básicas optimizables.
- Cuando un comando que puede beneficiarse de Rushmore contiene una cláusula WHILE.
- Cuando se disponga de poca memoria. La recuperación de datos sigue su curso, pero no se optimiza.

Desactivar Rushmore

Aunque no deseará hacerlo a menudo, es posible desactivar Rushmore. Cuando ejecute un comando que utilice Rushmore, Visual FoxPro determinará inmediatamente qué registros coinciden con la expresión de la cláusula FOR. A continuación, el comando manipula estos registros.

Si un comando que se puede optimizar modifica la clave de índice en la cláusula FOR, el conjunto de registros sobre el que opera Rushmore puede quedarse desfasado. En este caso, puede desactivar Rushmore para asegurarse de que disponga de la información más actualizada de la tabla.

Para desactivar Rushmore para un comando individual

- Utilice la cláusula NOOPTIMIZE.

Por ejemplo, este comando LOCATE no está optimizado:

```
LOCATE FOR DueDate < {^1998-01-01} NOOPTIMIZE
```

Puede desactivar o activar globalmente Rushmore para todos los comandos que se benefician de Rushmore, con el comando SET OPTIMIZE.

Para desactivar Rushmore globalmente

- Utilice el código siguiente:

```
SET OPTIMIZE OFF
```

Para desactivar Rushmore globalmente

- Utilice el código siguiente:

```
SET OPTIMIZE ON
```

El valor predeterminado de la optimización Rushmore está definido como ON.

Optimizar expresiones Rushmore

La tecnología Rushmore depende de la presencia de una *expresión básica optimizable* en una cláusula FOR. Una expresión básica optimizable puede formar una expresión completa o puede aparecer como parte de una expresión. También puede combinar expresiones básicas para formar una expresión optimizable compleja.

Crear expresiones básicas optimizables

Una expresión básica optimizable toma una de las dos formas siguientes:

eIndex relOp eExp

–O bien–

eExpr relOp eIndex

Una expresión básica optimizable tiene las siguientes características:

- *eIndex* coincide exactamente con la expresión sobre la cual está construido un índice.
- *eExp* es cualquier expresión y puede incluir variables y campos de otras tablas no relacionadas.
- *OpRel* es uno de los siguientes operadores: <, >, =, <=, >=, <>, #, ==, o !=. También puede utilizar las funciones ISNULL( ), BETWEEN( ) o INLIST( ) (o sus equivalentes SQL , como IS NULL, etc.).

Puede utilizar BETWEEN( ) o INLIST( ) de las dos formas siguientes:

eIndex BETWEEN(*eIndex*, *eExpr*, *eExpr*)

–O bien–

eExpr INLIST(*eIndex*, *eExpr*)

Nota ISBLANK( ) y EMPTY( ) no son optimizables con la tecnología Rushmore.

Si crea los índices `firstname`, `custno`, `UPPER(lastname` y `hiredate`, cada una de las siguientes expresiones es optimizable:

```
firstname = "Carlos"
custno >= 1000
UPPER(lastname) = "Martín"
hiredate < {^1997-12-30}
```

Una expresión optimizable puede contener variables y funciones que se evalúan como un valor concreto. Por ejemplo, al usar el índice `addr`, si ejecuta el comando "`WASHINGTON AVENUE" TO cVar`, las siguientes instrucciones también serían expresiones básicas optimizables:

```
ADDR = cVar
ADDR = SUBSTR(cVar,8,3)
```

Cuándo se optimizan las consultas

Es importante entender cuándo se optimizarán las consultas y cuándo no. Visual FoxPro optimiza las condiciones de búsqueda; para ello, busca una coincidencia exacta entre el lado izquierdo de una expresión de filtro y una expresión de clave de índice. Por tanto, Rushmore puede optimizar una expresión sólo si busca la expresión exacta usada en un índice.

Por ejemplo, imagine que acaba de crear una tabla y va a agregar el primer índice mediante un comando como el siguiente:

```
USE CUSTOMERS
INDEX ON UPPER(cu_name) TAG name
```

El comando siguiente no es optimizable porque la condición de búsqueda se basa únicamente en el campo `cu_name` y no en una expresión que esté indexada:

```
SELECT * FROM customers WHERE cu_name ="ACME"
```

En su lugar, debería crear una expresión optimizable con un comando como el siguiente, en la que la expresión que se esté buscando coincida exactamente con la expresión indexada:

```
SELECT * FROM customers WHERE UPPER(cu_name) = "ACME"
```

Sugerencia Para determinar el nivel de optimización Rushmore utilizado, llame a SYS(3054).

Combinar expresiones básicas optimizables

Puede combinar expresiones simples o complejas basadas en las cláusulas FOR o WHERE para incrementar la velocidad de recuperación de datos. Esto es posible si las expresiones FOR tienen las características de las expresiones básicas optimizables.

Las expresiones básicas pueden ser optimizables. Puede combinar expresiones básicas con los operadores lógicos AND, OR y NOT para formar una expresión de cláusula FOR compleja que también se puede optimizar. Una expresión creada con una combinación de expresiones básicas optimizables es totalmente optimizable. Si una o más de las expresiones básicas no son optimizables, la expresión compleja se podría optimizar parcialmente o bien no ser optimizable en absoluto.

Un conjunto de reglas determina si una expresión formada por expresiones básicas optimizables o no optimizables se puede optimizar totalmente, parcialmente o no se puede optimizar. La tabla siguiente resume las reglas de optimización de consultas de Rushmore.

Combinar expresiones básicas

Expresión básica	**Operador**	**Expresión básica**	**Resultados de la consulta**
Optimizable	AND	Optimizable	Totalmente optimizable
Optimizable	OR	Optimizable	Totalmente optimizable
Optimizable	AND	No optimizable	Parcialmente optimizable
Optimizable	OR	No optimizable	No optimizable
No optimizable	AND	No optimizable	No optimizable
No optimizable	OR	No optimizable	No optimizable
—	NOT	Optimizable	Totalmente optimizable
—	NOT	No optimizable	No optimizable

Puede utilizar el operador AND para combinar dos expresiones optimizables en una expresión totalmente optimizable:

```
FIRSTNAME = "CARLOS" AND HIREDATE < {^1997-12-30}      && Optimizable
```

En el siguiente ejemplo, el operador OR combina una expresión básica optimizable con una expresión no optimizable para crear una expresión que no es optimizable:

```
FIRSTNAME = "CARLOS" OR "S" $ LASTNAME      && No optimizable
```

Puede crear una expresión totalmente optimizable si utiliza el operador NOT con una expresión optimizable:

```
NOT FIRSTNAME = "CARLOS"      && Totalmente optimizable
```

También puede utilizar paréntesis para agrupar combinaciones de expresiones básicas.

Combinar expresiones complejas

Del mismo modo que puede combinar expresiones básicas, puede combinar expresiones complejas para crear una expresión aún más compleja totalmente optimizable, parcialmente optimizable o no optimizable. A su vez, puede combinar estas expresiones más complejas para crear expresiones que se pueden optimizar total o parcialmente o que no se pueden optimizar. La tabla siguiente describe los resultados de combinar estas expresiones complejas. Estas reglas también se aplican a expresiones agrupadas con paréntesis.

Combinación de expresiones complejas

Expresión	Operador	Expresión	Resultado
Totalmente optimizable	AND	Totalmente optimizable	Totalmente optimizable
Totalmente optimizable	OR	Totalmente optimizable	Totalmente optimizable
Totalmente optimizable	AND	Parcialmente optimizable	Parcialmente optimizable
Totalmente optimizable	OR	Parcialmente optimizable	Parcialmente optimizable
Totalmente optimizable	AND	No optimizable	Parcialmente optimizable
Totalmente optimizable	OR	No optimizable	No optimizable
—	NOT	Totalmente optimizable	Totalmente optimizable
Parcialmente optimizable	AND	Parcialmente optimizable	Parcialmente optimizable
Parcialmente optimizable	OR	Parcialmente optimizable	Parcialmente optimizable
Parcialmente optimizable	AND	No optimizable	Parcialmente optimizable
Parcialmente optimizable	OR	No optimizable	No optimizable
—	NOT	Parcialmente optimizable	No optimizable
No optimizable	AND	No optimizable	No optimizable
No optimizable	OR	No optimizable	No optimizable

—	NOT	No optimizable	No optimizable

Puede combinar expresiones totalmente optimizables con el operador OR para crear una expresión que también sea totalmente optimizable:

```
* Expresión totalmente optimizable
(FIRSTNAME = "CARLOS" AND HIREDATE < {^1997-12-30}) ;
   OR (LASTNAME = "" AND HIREDATE > {^1996-12-30})
```

Para crear expresiones parcialmente optimizables, combine una expresión totalmente optimizable con una expresión que no sea optimizable. En el siguiente ejemplo, el operador AND se utiliza para combinar estas expresiones:

```
* Expresión parcialmente optimizable
(FIRSTNAME = "FRED" AND HIREDATE < {^1997-12-30}) ;
   AND "S" $ LASTNAME
```

Las expresiones parcialmente optimizables se pueden combinar para crear una expresión que también sea parcialmente optimizable:

```
* Expresión parcialmente optimizable
(FIRSTNAME = "CARLOS" AND "S" $ LASTNAME) ;
   OR (FIRSTNAME = "PEDRO" AND "T" $ LASTNAME)
```

La combinación de expresiones que no sean optimizables crea una expresión que tampoco se puede optimizar:

```
* Expresión que no es optimizable
("CARLOS" $ FIRSTNAME OR "S" $ LASTNAME) ;
   OR ("MAIN" $ STREET OR "AVE" $ STREET)
```

Optimizar formularios y controles

También puede mejorar de forma significativa los formularios y controles de la aplicación.

Sugerencia Si desea más información sobre el establecimiento y obtención de propiedades de forma eficaz, vea Referencias a propiedades de objetos de forma eficaz, más adelante en este capítulo.

Usar el entorno de datos

Si utiliza el entorno de datos del Diseñador de formularios o el Diseñador de informes, el rendimiento de la apertura de la tabla es mucho más rápido que si se ejecutaran los comandos USE, SET ORDER y SET RELATION en el evento Load del formulario. Cuando use el entorno de datos, Visual FoxPro utiliza las llamadas al motor de nivel inferior para abrir las tablas y configurar los índices y las relaciones.

Limitar el número de formularios en un conjunto de formularios

Utilice conjuntos de formularios solamente cuando sea necesario que un grupo de formularios comparta una sesión de datos privada. Cuando utilice un conjunto de formularios, Visual FoxPro crea

instancias de todos los formularios y de todos los controles de todos los formularios del conjunto, aun cuando sólo se muestre en pantalla el primer formulario del conjunto. Esto puede resultar lento e innecesario si los formularios no tienen que compartir una sesión de datos privada. En su lugar, debería ejecutar DO FORM para los otros formularios cuando sean necesarios.

Sin embargo, si utiliza un conjunto de formularios, obtendrá una pequeña mejora del rendimiento cuando tenga acceso a los formularios del conjunto porque los formularios ya estarán cargados aunque no visibles.

Carga dinámica de controles de página en un marco de páginas

Los marcos de página, al igual que los conjuntos de formularios, cargan todos los controles de cada página en el momento de cargar el marco de páginas lo que da lugar a una demora perceptible cuando se carga este último. En su lugar, podría cargar los controles de página dinámicamente a medida que se fueran necesitando, creando una clase fuera de los controles de cada página y cargándolos cuando se activara la página.

Para cargar dinámicamente los controles de página

1. Diseñe el formulario de la forma habitual, incluyendo todos los controles en todas las páginas.

2. Una vez terminado el diseño, vaya a la segunda página del marco de página y guarde los controles allí existentes como una clase.

3. Abra la clase que acaba de crear y compruebe que los controles sigan dispuestos de la forma correcta.

4. Repita los pasos 2 y 3 para la tercera y las páginas siguientes del marco de página.

5. En el evento Activate de la segunda página y de las siguientes del marco de página, agregue objetos y déjelos visibles.

   Por ejemplo, si la clase de controles se llama `cnrpage1`, debe agregar el código siguiente:

   ```
   IF THIS.ControlCount = 0
   THIS.AddObject("cnrpage1","cnrpage1")
   THIS.cnrpage1.Visible = .T.
   ENDIF
   ```

Vinculación dinámica de controles a datos

Puede agilizar el tiempo de carga de un formulario que contenga numerosos controles vinculados a datos si retrasa la vinculación de esos controles hasta el momento en que sean necesarios.

Para vincular dinámicamente controles a datos

1. Sitúe las tablas y vistas del formulario en el entorno de datos de modo que se abran cuando se cargue el formulario.

2. Para cada control dependiente, agregue código al código de evento GotFocus que vincula el

control al valor de los datos. Por ejemplo, el código siguiente vincula un control ComboBox al campo `customer.company`:

```
* Comprobar si ya se ha vinculado el control.
IF THIS.RecordSource = ""
* Establecer el valor del origen de registros
* y establecer el tipo de origen de registros como "fields"
THIS.RecordSource = "customer.company"
THIS.RecordSourceType = 6
THIS.Refresh
ENDIF
```

Retardo de la actualización de pantalla

Si realiza varios cambios en la pantalla, por ejemplo, cambia los valores de varios controles a la vez, puede reducir el tiempo global necesario para actualizar la pantalla si retarda la actualización de pantalla hasta que se realicen todos los cambios. Por ejemplo, si hace que los controles queden visibles o invisibles, cambia los colores de los controles o mueve los registros de controles dependientes, resulta mucho más eficaz retardar el relleno de color de estos controles hasta que se hayan completado todos los cambios:

Para retardar la actualización de la pantalla

1. Establezca la propiedad LockScreen del formulario como verdadera.
2. Actualice los controles cuando sea necesario.
3. Llame al método Refresh del formulario.
4. Establezca la propiedad LockScreen del formulario como falsa.

El ejemplo siguiente cambia varias propiedades de la pantalla a la vez, se desplaza a un registro nuevo y sólo entonces actualiza la pantalla con información nueva. Si LockScreen no se hubiera establecido como verdadero, en cada una de estas operaciones se volverían a dibujar los controles afectados individualmente y el rendimiento global de la actualización parecería retardado.

```
THISFORM.LockScreen = .T.
THISFORM.MyButton.Caption = "Guardar"
THISFORM.MyGrid.BackColor = RGB (255, 0, 0)   && Rojo
SKIP IN customers
SKIP IN orders
THISFORM.Refresh
THISFORM.LockScreen = .F.
```

Sugerencia Esta técnica no proporciona ninguna ventaja si está actualizando un único control.

Reducir código en métodos usados con frecuencia

Puesto que el método Refresh y el evento Paint suelen utilizarse con frecuencia, puede aumentar el rendimiento de formularios si reduce la cantidad de código de estos métodos. De forma análoga, para acelerar el tiempo de carga de un formulario, podría mover código desde el evento Init a los métodos usados con menos frecuencia como Activate, Click y GotFocus. Entonces, use una propiedad del

control (como Tag o una propiedad personalizada) para hacer un seguimiento de si el control ya ha ejecutado el código que sólo necesita ejecutarse una vez.

Optimizar programas

Escribiendo código cuidadosamente, puede escribir programas más rápidos. Son varias las maneras de aumentar el rendimiento de los programas en Visual FoxPro:

- Seguir las sugerencias generales sobre el rendimiento de programación que aparecen a continuación.
- Usar expresiones de nombres en lugar de la sustitución mediante macros.
- Hacer referencia a propiedades de objetos de forma eficaz.

Sugerencias generales sobre el rendimiento

Para escribir programas más rápidos, siga las recomendaciones que se indican a continuación.

- Elija el tipo de datos correcto. En particular, use el tipo de datos Integer para la información numérica cuando sea posible, puesto que se procesa de forma más eficaz. Siempre que sea posible, use el tipo de datos Integer para los valores de claves externas, lo que dará lugar a archivos de datos e índices más pequeños (y, por tanto, más rápidos) y a combinaciones más rápidas.

  Nota Para ver un ejemplo que muestre la forma de crear un índice más pequeño y, por tanto, más rápido, ejecute Solution.app en el directorio ...\Samples\Vfp98\Solution de Visual Studio. Elija **Ver los ejemplos mediante una lista filtrada**, seleccione **Índice** en la lista desplegable y después elija **Crear índices pequeños usando BINTOC( )** en la lista que aparece.

- Evite volver a abrir archivos, puesto que esto ralentiza el rendimiento. En su lugar, asigne archivos a áreas de trabajo cuando los abra y después use el comando SELECT para elegir un área de trabajo específica cuando se precise.
- Los bucles FOR … ENDFOR son más rápidos que los bucles DO WHILE … ENDDO.
- Cuando copie datos de varios campos, SCATTER TO ARRAY es más rápido que SCATTER MEMVAR.
- Para optimizar el uso de la memoria, evite la creación de objetos antes de necesitarlos y borre los objetos cuando deje de trabajar con ellos para liberar memoria.

  Sugerencia Puede comprobar cuánta memoria consume cada objeto si llama a la función SYS (1016).

- Envíe la salida a la ventana superior siempre que sea posible; la actualización de ventanas situadas detrás de la superior es bastante más lenta. No se debe, en ningún caso, hacer que la salida se desplace detrás de una ventana.
- Desactive la presentación del estado con el comando SET TALK OFF, que elimina el trabajo de la actualización de la pantalla.
- Establezca el comando SET DOHISTORY como OFF para evitar la actualización de la ventana de comandos cada vez que se ejecute un programa.

Usar expresiones de nombre en lugar de sustitución de macros

Si utiliza expresiones de nombre en lugar de la sustitución de macros, el rendimiento del programa aumentará notablemente. Por ejemplo, si asigna un valor a la variable `cFile`, una expresión de nombre creada con cFile será más rápida que la sustitución de macros.

```
cFile = "CUST"
use &cFile      && Substitución de macros, lenta
use (cFile)      && Expresión de nombre: rápida, preferida
```

Referencias a propiedades de objetos de forma eficaz

Comprendiendo la forma en que funciona Visual FoxPro con las propiedades y objetos, puede optimizar la ejecución de las aplicaciones.

Optimizar referencias a una propiedad repetidas

Cuando haga referencia a una propiedad de objeto con la sintaxis *objeto.propiedad*, Visual FoxPro debe buscar el objeto antes de poder tener acceso a la propiedad. Si debe tener acceso a la propiedad repetidas veces, esta estrategia de búsqueda puede ralentizar el rendimiento.

Para evitar las referencia al mismo procedimiento varias veces (como en un bucle), lea el valor de la propiedad en una variable, realice los cambios y establezca la propiedad una sola vez cuando termine. Por ejemplo, el código siguiente rellena una matriz de propiedades; para ello, crea primero una matriz en la memoria, la rellena y después establece la propiedad una sola vez al final:

```
* Copiar cadena a una variable localvariable
lcChar = THISFORM.cCharString
LOCAL laCharArray[256]   && Crear matriz local
FOR nCounter = 1 to 256
   laCharArray[x] = SUBSTR(laChar,x,1)
ENDFOR
* Copiar la matriz local a la matriz de propiedades
ACOPY(laCharArray,THISFORM.aCharArray)
```

Hacer referencia a múltiples propiedades de forma eficaz

Si actualiza más de una propiedad para el objeto, Visual FoxPro debe buscar el objeto varias veces, lo que puede afectar al rendimiento. En el ejemplo siguiente, el código hace que Visual FoxPro busque en cuatro objetos (como `THISFORM`, `pgfCstInfo`, `pgCstName` y `txtName`) para encontrar la propiedad que se va a establecer. Dado que el código establece dos propiedades, la búsqueda a cuatro niveles se realiza dos veces:

```
THISFORM.pgfCstInfo.pgCstName.txtName.Value = ;
   "Carlos Martín"
THISFORM.pgfCstInfo.pgCstName.txtName.BackColor = ;
   RGB (0,0,0)  & Rojo oscuro
```

Para evitar este volumen de trabajo, use el comando WITH … ENDWITH. Este método hace que Visual FoxPro busque el objeto una sola vez. Por ejemplo, el ejemplo siguiente realiza la misma tarea que el anterior, pero de forma más rápida:

```
WITH THISFORM.pgfCstInfo.pgCstName.txtName
   .Value = "Carlos Martín"
   .BackColor = RGB (0,0,0)  & Rojo oscuro
ENDWITH
```

También puede almacenar una referencia a un objeto en una variable y después incluir la variable en lugar de la referencia al objeto:

```
oControl = THISFORM.pgfCstInfo.pgCstName.txtName
oControl.Value = "Carlos Martín"
oControl.BackColor = RGB (0,0,0)  & Rojo oscuro
```

Optimizar controles ActiveX

Si utiliza Automatización o controles ActiveX en la aplicación, puede ajustar la aplicación para obtener el máximo rendimiento de los controles ActiveX y de Automatización.

Usar controles ActiveX de forma eficaz

Para obtener el máximo rendimiento a la hora de utilizar controles ActiveX en los formularios, siga estas recomendaciones:

- Inicie los servidores de Automatización en primer lugar. Los controles enlazados a campos generales suelen funcionar mejor cuando los servidores de estos tipos de datos (como Microsoft Excel o Word) ya se están ejecutando en el equipo cliente.
- Inserte objetos "como un icono". Cuando inserte un control ActiveX en un campo, hágalo como un icono o un marcador de posición en lugar de insertar el objeto completo. Esto reduce la cantidad de espacio de almacenamiento necesario porque Visual FoxPro almacena una imagen de presentación con el objeto, lo que consume gran cantidad de espacio de almacenamiento. Si se inserta un objeto como un icono también se aumenta el rendimiento para dibujar el objeto.
- Use controles de imágenes. Si desea mostrar un mapa de bits (como, el logotipo de una empresa), los controles de imágenes son mucho más rápidos que los controles OLEBound.
- Use vínculos manuales siempre que sea posible. Los vínculos manuales a los objetos son más rápidos porque evitan el tiempo de notificación requerido para los vínculos automáticos y porque el servidor no necesita iniciarse para dibujar el objeto. Si no necesita actualizar un objeto con frecuencia, use vínculos manuales.

Optimizar el rendimiento de Automatización

Si la aplicación interactúa con otras aplicaciones, puede conseguir el máximo rendimiento mediante las técnicas siguientes.

Evitar múltiples instancias del servidor

En algunos casos, los servidores de Automatización (como Microsoft Excel) iniciarán una nueva instancia, aun cuando ya haya una ejecutándose. Para evitar esto y aumentar el rendimiento, use la función GetObject( ) en lugar de CreateObject( ). Por ejemplo, la llamada siguiente siempre utilizará una instancia existente:

```
x = GetObject(,"excel.Application")
```

De lo contrario, la llamada siguiente creará una instancia nueva:

```
x = CreateObject("excel.Application")
```

Si llama a GetObject( ) pero el servidor no está funcionando, aparecerá el error 1426. En ese caso, puede interrumpir el error y llamar a CreateObject( ):

```
ON ERROR DO oleErr WITH ERROR()
x = GetObject(,"excel.application")
ON ERROR  && restablecer el controlador de errores del sistema

PROCEDURE oleErr
PARAMETER mError
IF mError = 1426 then
   x = CreateObject("excel.application")
ENDIF
```

Hacer referencia a objetos de forma eficaz

La ejecución de expresiones que usen objetos en un servidor de Automatización puede resultar muy costosa, especialmente cuando se evalúan varias veces. Es mucho más económico almacenar las referencias de objetos en variables para tenerlas como referencia. Para obtener más información, vea Optimizar referencias repetidas a una propiedad, en este mismo capítulo.

Optimizar aplicaciones en entornos multiusuario

Si va a escribir aplicaciones para un entorno multiusuario, el rendimiento es un factor especialmente importante, porque las operaciones ineficaces se multiplican. Además, si varios usuarios tienen acceso a datos, la aplicación debe gestionar problemas de simultaneidad y acceso a la red.

Para administrar estos problemas, puede:

- Ajustar el intervalo de reintentos de bloqueo.
- Usar el proceso de transacciones de forma eficaz.

Asimismo, las sugerencias para trabajar con datos almacenados en servidores remotos pueden resultarle especialmente útiles. Para obtener más información, vea Optimizar el acceso a datos remotos, más adelante en este capítulo.

Ajustar el intervalo de reintentos de bloqueo

Si la aplicación intenta bloquear un registro o una tabla sin éxito, puede hacer que Visual FoxPro reintente el bloqueo automáticamente después de un breve intervalo. Sin embargo, cada intento de bloqueo da lugar a un tráfico de red más intenso. Si ya fuera intenso, el envío de repetidas solicitudes de bloqueo agregará una sobrecarga de trabajo a la red y provocará una ralentización global para todos los usuarios.

Para controlar esta situación, puede ajustar el intervalo entre los intentos de bloqueo. Si usa un

intervalo más grande (lo que provocaría menos reintentos por segundo) se reducirá el tráfico de la red y se aumentará el rendimiento.

Para ajustar el intervalo de reintentos de bloqueo

- Llame a la función SYS(3051); para ello, indique el número de milisegundos que debe esperar entre cada intento de bloqueo.

Usar el procesamiento de transacciones de forma eficaz

Cuando utiliza el procesamiento de transacciones, debe diseñar las transacciones de forma que se reduzca al mínimo el efecto que puedan tener en otros usuarios. Mientras esté abierta una transacción, todos los bloqueos establecidos durante la transacción permanecerán bloqueados hasta que se confirmen o se deshagan. Aun cuando ejecute un comando UNLOCK explícito, los bloqueos se mantendrán hasta que ejecute el comando END TRANSACTION o ROLLBACK.

Además si anexa registros a una tabla, es imprescindible que Visual FoxPro bloquee el encabezado de la tabla. El encabezado permanece bloqueado durante el proceso de la transacción, lo que impide que los demás usuarios también puedan anexar registros.

Para reducir al mínimo el impacto de las transacciones, diséñelas de modo que comiencen y terminen lo más cerca posible de la actualización real de los datos; la transacción ideal sólo contiene instrucciones de actualización de datos.

Si va a agregar el proceso de transacciones a actualizaciones de datos realizadas en un formulario, no abra una transacción, ejecute el formulario y después confirme la transacción cuando el formulario esté cerrado. En su lugar, ponga las instrucciones de proceso de transacciones en el código de eventos para el botón Guardar, por ejemplo:

```
* Guardar método del botón de comando cmdSave
BEGIN TRANSACTION
UPDATE PRODUCTS SET reorder_amt = 0 WHERE discontinued = .T.
END TRANSACTION
```

Optimizar el acceso a datos remotos

La recuperación de datos de cualquier base de datos remota resulta cara. Con el fin de obtener datos de una base de datos del servidor, hay que seguir estos pasos:

1. El cliente ejecuta la consulta en la base de datos remota.
2. El servidor analiza y compila la consulta.
3. El servidor genera un conjunto de resultados.
4. El servidor indica al cliente que se ha completado el resultado.
5. El cliente recoge los datos del servidor a través de la red. Este paso puede realizarse en una sola operación o el cliente puede solicitar que los resultados se envíen en partes a medida que se

vayan necesitando.

Puede usar varias técnicas para aumentar la velocidad de la recuperación o actualización de los datos. En la sección siguiente se tratan estas estrategias:

- Recuperar sólo los datos necesarios
- Actualizar tablas remotas de forma eficaz
- Enviar instrucciones en un lote
- Establecer el tamaño del paquete
- Retrasar la recuperación de datos memo y binarios
- Almacenar localmente datos de consulta
- Crear reglas locales

Obtener sólo los datos necesarios

La mayor parte de las aplicaciones que usan datos remotos, formularios e informes no necesitan tener acceso a todos los datos de una tabla a la vez. Por tanto, puede aumentar el rendimiento si crea vistas remotas que busquen o actualicen únicamente los campos y registros que desee con lo que se reduce al mínimo la cantidad de datos que deben transmitirse a través de la red.

Para crear consultas que minimicen el trabajo de recuperación de datos de orígenes remotos, siga estas sugerencias:

- Especifique sólo los campos que necesite. No use la instrucción `SELECT * FROM customers` a menos que necesite todos los campos de la tabla.
- Incluya una cláusula WHERE para limitar el número de registros transferidos. Cuanto más específica sea la cláusula WHERE, menos registros se transmitirán al equipo y con más rapidez se terminará la consulta.
- Si no puede predecir durante el diseño los valores que se van a utilizar en una cláusula WHERE, puede utilizar parámetros en la cláusula. Cuando se ejecute la consulta, Visual FoxPro usará el valor de una variable de parámetro o solicitará al usuario el valor de búsqueda. Por ejemplo, esta consulta permite a la aplicación o al usuario rellenar la región en el tiempo de ejecución:

  ```
  SELECT cust_id, company, contact, address ;
  FROM customers ;
  WHERE region = ?pcRegion
  ```

- Establezca la propiedad NoDataOnLoad del objeto Cursor en el entorno de datos correspondiente. Esta técnica se suele utilizar con vistas parametrizadas en las que los datos del parámetro proceden del valor de un control de un formulario.

Actualizar tablas remotas de forma eficaz

Cuando se usa una vista para actualizar una tabla en un origen de datos remoto, Visual FoxPro debe comprobar si los registros que se están actualizando han sufrido alguna modificación. Para ello, Visual FoxPro debe examinar los datos en el servidor y compararlos con los datos existentes en su equipo. En algunos casos, esta operación puede resultar lenta.

Para optimizar el proceso de actualización de datos en orígenes de datos remotos, puede especificar la forma en que Visual FoxPro debe comprobar los registros modificados. Para ello, tiene que indicar la cláusula WHERE que Visual FoxPro debe generar para realizar la actualización.

Imagine, por ejemplo, que está usando una vista basada en una tabla de clientes en un origen de datos remoto. Ha creado la vista mediante una instrucción SELECT - SQL como la siguiente:

```
SELECT cust_id, company, address, contact ;
   FROM customers ;
   WHERE region = ?vpRegion
```

Quiere actualizar los cuatro campos que ha especificado en la vista, salvo el campo clave (cust_id). En la tabla siguiente se presenta la cláusula WHERE que Visual FoxPro generará para cada una de las opciones disponibles en la cláusula SQL WHERE.

Nota La función OLDVAL( ) devuelve la versión preactualizada de los campos que se han modificado y la función CURVAL( ) devuelve el valor actual almacenado en el origen de datos remoto. Si compara estos valores, Visual FoxPro puede determinar si el registro ha cambiado en el origen de datos remoto desde que se transfirió al equipo.

Valor	Cláusula WHERE resultante
Sólo campos clave	WHERE OLDVAL(cust_id) = CURVAL(cust_id)
Campos clave y actualizables (predeterminado)	WHERE OLDVAL(cust_id) = CURVAL(cust_id) AND OLDVAL(<mod_fld1>) = CURVAL(<mod_fld2>) AND OLDVAL(<mod_fld2>) = CURVAL(<mod_fld2>) AND ...
Campos clave y modificados	WHERE OLDVAL(cust_id) = CURVAL(cust_id) AND OLDVAL(company) = CURVAL(company) AND OLDVAL(contact) = CURVAL(contact) AND OLDVAL(address) = CURVAL(address)
Clave y marca de hora	WHERE OLDVAL(cust_id) = CURVAL(cust_id) AND OLDVAL(timestamp) = CURVAL(timestamp)

En general, debería elegir una opción para la cláusula SQL WHERE en este orden de preferencia:

1. **Clave y marca de hora**, si la base de datos remota admite los campos de marca de hora, que es la forma más rápida de indicar si se ha modificado algún registro.

2. **Campos clave y modificados**, porque los campos que se actualizan en el servidor son casi siempre un subconjunto del número total de campos que se podría actualizar.

3. **Campos clave y actualizables**.

4. **Sólo campos clave**. El uso de esta configuración implica que el servidor remoto insertará un

registro totalmente nuevo que use la clave modificada y eliminará el registro anterior.

Enviar instrucciones en un lote

Algunos servidores (como Microsoft SQL Server) permiten enviar un lote de instrucciones SQL en un solo paquete. Esto aumenta el rendimiento porque se reduce el tráfico de la red y porque el servidor puede compilar múltiples instrucciones a la vez.

Por ejemplo, si especifica un tamaño de lote de cuatro y actualiza 10 registros en una base de datos, Visual FoxPro envía en un solo lote cuatro instrucciones como la siguiente a la base de datos del servidor:

```
UPDATE customer SET contact = "Agustina Rivera" ;
   WHERE cust_id = 1;
UPDATE customer SET contact = "Cristina Martínez" ;
   WHERE cust_id = 2;
UPDATE customer SET company = "Enrique Ballina" ;
   WHERE cust_id = 3;
UPDATE customer SET contact = "Ernesto Méndez" ;
   WHERE cust_id = 4
```

Para enviar instrucciones en un lote

- En el cuadro de diálogo **Opciones**, elija la ficha **Datos remotos** y en **Registros para actualizar por lotes** especifique el número de registros que van a incluirse en el lote.

  –O bien–

- Llame a las funciones DBSETPROP( ) o CURSORSETPROP( ) para establecer estas propiedades:
  - Establezca Transaction a 2.
  - Establezca BatchUpdateCount al número de instrucciones que se van a enviar en un lote.

  –O bien–

1. En el **Diseñador de vistas**, elija **Opciones avanzadas** del menú **Consulta** para abrir el cuadro de diálogo **Opciones avanzadas**.
2. En el área **Rendimiento**, situada junto a **Número de registros para actualizar por lotes**, especifique el número de instrucciones que se van a enviar en un lote.

   Nota Debería probar con diferentes valores para esta propiedad y la propiedad PacketSize para optimizar las actualizaciones.

Establecer el tamaño del paquete

Puede optimizar el acceso a servidores remotos; para ello, ajuste el tamaño del paquete de red que se envía y se obtiene de la base de datos remota. Por ejemplo, si la red admite tamaños grandes (más de 4096 bytes), puede aumentar el tamaño del paquete en Visual FoxPro con el fin de enviar más datos

cada vez que lea o escriba en la red.

Para establecer el tamaño del paquete

- Llame a las funciones DBSETPROP( ) o CURSORSETPROP( ) y establezca la propiedad PacketSize a un valor entero positivo. El valor predeterminado es 4096.

  Nota Es posible que distintos proveedores de red administren esta propiedad de forma distinta por lo que deberá consultar la documentación de su servicio de red. Novell® NetWare®, por ejemplo, tiene un tamaño de paquete máximo de 512 bytes por lo que si se establece la propiedad PacketSize a un valor superior a éste no se obtendrá ninguna ventaja adicional.

Retardo de la recuperación de datos memo y binarios

Si está almacenando datos memo o binarios en un servidor remoto, puede aumentar el rendimiento; para ello, retarde la transferencia de estos datos hasta que la aplicación la requiera realmente.

Para retardar la recuperación de los datos memo y binarios

- En el cuadro de diálogo **Opciones**, elija la ficha **Datos remotos** y en **Opciones predeterminadas de vista remota**, establezca **Buscar memo**.

  –O bien–

- Llame a las funciones DBSETPROP( ) o CURSORSETPROP( ) para establecer la propiedad FetchMemo.

Almacenamiento local de datos de consulta

Muchas aplicaciones incluyen datos de consulta estáticos, como abreviaturas de estados, códigos postales y cargos de empleados. Si la aplicación contiene este tipo de datos y si la tabla no es demasiado grande, podría aumentar la velocidad de la aplicación; para ello, mantenga copias de esta información en el equipo de cada usuario, ya que las consultas no generan tráfico de red.

Esta técnica es especialmente útil para los datos que nunca cambian o lo hacen muy esporádicamente. Si los datos cambian en alguna ocasión, debe diseñar una estrategia para transferir una copia nueva de la tabla de consultas al equipo de cada usuario.

Crear reglas locales

Puede aumentar el rendimiento de la aplicación si crea reglas locales a nivel de campo y de registro en Visual FoxPro, en lugar de utilizar las reglas definidas en el servidor. Estas reglas pueden impedir que los datos que no son compatibles con las reglas de datos o de la empresa se introduzcan en la base de datos.

Mediante la definición de reglas en Visual FoxPro, se detectan los datos no válidos antes de enviarse a través de la red, lo que es más rápido y ofrece un mejor control para administrar las condiciones de error. Sin embargo, el uso de reglas locales también implica su coordinación con las reglas del

servidor remoto. Por ejemplo, si se modifican las reglas del servidor, tendría que cambiar las reglas locales para que coincidieran.

Para obtener información sobre la creación de reglas locales, vea la sección Actualizar datos en una vista del capítulo 8, "Crear vistas".

Optimizar aplicaciones internacionales

Si va a programar aplicaciones internacionales, podría necesitar administrar la secuencia de ordenación de los datos para obtener el máximo rendimiento. Esta sección trata sobre lo siguiente:

- Usar secuencias de ordenación de forma eficaz.
- Usar SELECT - SQL con múltiples secuencias de ordenación.

Usar secuencias de ordenación de forma eficaz

Si los datos no incluyen marcas diacríticas, como acentos (á) o diéresis (ü), puede aumentar el rendimiento mediante la secuencia de ordenación del equipo porque:

- Las claves de índices que no son del equipo son dos veces más grandes puesto que contienen la información diacrítica.
- La ordenación que no es del equipo usa muchas reglas especiales para indexar caracteres con el fin de que devuelvan los resultados adecuados.

Dado que la secuencia de ordenación del equipo es más rápida, se suele preferir para combinaciones y búsquedas mientras que otras secuencias de ordenación resultan perfectas para ordenar registros.

Cuando cree un índice, Visual FoxPro usa el valor actual de SET COLLATE. Por tanto, si desea crear dos índices con dos secuencias de ordenación, puede usar una secuencia de comandos como la siguiente:

```
SET COLLATE TO "MACHINE"
INDEX ON lastname TAG _lastname     && combinar/buscar índice
SET COLLATE TO "GENERAL"
INDEX ON lastname TAG lastname  && índice de ordenación
```

Cuando desee buscar, seleccionar o combinar el campo `lastname`, ejecute el comando SET COLLATE TO "MACHINE" antes de realizar la operación. Rushmore utilizará el índice creado en la secuencia de ordenación del equipo y la operación de búsqueda será muy rápida.

Usar SQL SELECT con múltiples secuencias de ordenación

Cuando ejecute un comando SELECT - SQL, Visual FoxPro usará la secuencia de ordenación actual para realizar la búsqueda y para las cláusulas ORDER BY y GROUP BY. Si desea buscar y ordenar mediante diferentes secuencias de ordenación, puede dividir los comandos SQL en dos pasos, como se indica a continuación:

```
* Seleccionar registros usando una secuencia de ordenación
SET COLLATE TO "MACHINE"
SELECT * FROM table INTO CURSOR temp1 ;
```

```
    WHERE lname = "Alzaga"
* Ordenar registros usando una secuencia de ordenación diferente
SET COLLATE TO "GENERAL"
SELECT * FROM temp1 INTO TABLE output ORDER BY lastname
```